NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CROWDSOURCING PHYSICAL NETWORK TOPOLOGY MAPPING WITH NET.TAGGER

by

Daniel Glenn Woodman

March 2016

Thesis Advisor:	Robert Beverly
Second Reader:	Justin P. Rohrer

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	03-25-2016	Master's Thesis 09-29-2014 to 03-25-2016

4. TITLE AND SUBTITLE

CROWDSOURCING PHYSICAL NETWORK TOPOLOGY MAPPING WITH NET.TAGGER

5. FUNDING NUMBERS

N66001-2250-58231

6. AUTHOR(S)

Daniel Glenn Woodman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of Homeland Security
245 Murray Lane SW, Washington, DC 20528

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Despite significant research, the challenge of mapping the physical topology of large networks remains a relatively unsolved problem. Although it possesses numerous ramifications for Internet security and resiliency, physical network geolocation research has not matched corresponding advancements made in logical topology mapping. This thesis proposes net.Tagger: a novel approach to network infrastructure mapping that combines smartphone apps with crowdsourced collection to gather data for offline aggregation and analysis. The project aims to build a map of physical network infrastructure such as fiber-optic cables, facilities, and access points. The net.Tagger project aligns to the OpenStreetMap project, a proven, open-source framework for managing crowdsourced map data. This thesis delivers a working proof-of-concept system for further research, including a smartphone app for gathering physical topology data, and the backend services to process and store it. We also present the results of an initial release to 25 users, analysing collection trends and extrapolating to predict potential findings of a future large-scale release.

14. SUBJECT TERMS

Internet, network mapping, physical network topology, Internet backbone, crowdsourcing

15. NUMBER OF PAGES 111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CROWDSOURCING PHYSICAL NETWORK TOPOLOGY MAPPING WITH NET.TAGGER

Daniel Glenn Woodman
Lieutenant Junior Grade, United States Coast Guard
B.S., U.S. Coast Guard Academy, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2016

Author:	Daniel Glenn Woodman
Approved by:	Robert Beverly Thesis Advisor
	Justin P. Rohrer Second Reader
	Peter Denning Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Despite significant research, the challenge of mapping the physical topology of large networks remains a relatively unsolved problem. Although it possesses numerous ramifications for Internet security and resiliency, physical network geolocation research has not matched corresponding advancements made in logical topology mapping. This thesis proposes net.Tagger: a novel approach to network infrastructure mapping that combines smartphone apps with crowdsourced collection to gather data for offline aggregation and analysis. The project aims to build a map of physical network infrastructure such as fiber-optic cables, facilities, and access points. The net.Tagger project aligns to the OpenStreetMap project, a proven, open-source framework for managing crowdsourced map data. This thesis delivers a working proof-of-concept system for further research, including a smartphone app for gathering physical topology data, and the backend services to process and store it. We also present the results of an initial release to 25 users, analysing collection trends and extrapolating to predict potential findings of a future large-scale release.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**

1.1 Problem . . . 1

1.2 Research Question . . . 3

1.3 Contribution . . . 4

1.4 Thesis Organization . . . 4

2 Background **5**

2.1 Introduction . . . 5

2.2 Physical Internet Design . . . 5

2.3 Physical Topology Mapping History . . . 11

2.4 Crowdsourced Mapping . . . 15

2.5 Infrastructure Indicators . . . 18

2.6 Android Platform Capabilities . . . 27

3 Implementation **31**

3.1 Project Requirements. . . . 31

3.2 App Design . . . 34

3.3 Backend Services . . . 41

4 Testing and Results **45**

4.1 Initial Release . . . 45

4.2 Quality Examples . . . 52

4.3 Low-Permanence Indicators . . . 54

4.4 Tag Verification . . . 56

4.5 Tag Comments . . . 64

4.6 Errors and Noise . . . 65

5 Future Work **69**

5.1 App . . . 69

5.2 Server 75
5.3 Data Analysis 78
5.4 User Incentives 79

List of References **85**

Initial Distribution List **91**

List of Figures

Figure 2.1	Street Markings Color Code. Source: [39]	20
Figure 2.2	Orange Marking	21
Figure 2.3	Orange Marking	21
Figure 2.4	Duct Marking	22
Figure 2.5	Annotated Duct Marking	22
Figure 2.6	Bell System	23
Figure 2.7	US West	23
Figure 2.8	Communication Handhole	24
Figure 2.9	Computer Handhole	24
Figure 2.10	Fiber Optic 15/20K	24
Figure 2.11	SBC NewBasis 20K	24
Figure 2.12	Qwest Warning	25
Figure 2.13	Century Link Warning (Close-Up)	25
Figure 2.14	Cell Tower Markings	26
Figure 2.15	Hidden Cell Tower. Source: [40]	26
Figure 3.1	Initial Main Screen	34
Figure 3.2	Initial Submit Screen	34
Figure 3.3	Refined Main Screen	36
Figure 3.4	Refined Submit Screen	36
Figure 3.5	Examples Screen	40
Figure 4.1	CDF of Tags by User	47

Figure 4.2	CDF of Infrastructure Types by User	48
Figure 4.3	CDF of Infrastructure Providers by User	49
Figure 4.4	CDF of Zipcodes by User	50
Figure 4.5	CDF of Tagging Delay	51
Figure 4.6	Communications Vault with Duct	53
Figure 4.7	Duct with Building	54
Figure 4.8	Orange Marking and TV Pedestal, Bark and Grass	55
Figure 4.9	Duct Marking, Grass	55
Figure 4.10	User-submitted Image	56
Figure 4.11	Google Earth at Image Coordinates	57
Figure 4.12	Cell Tower, User Submitted	58
Figure 4.13	Cell Tower, Google Earth	59
Figure 4.14	Bell Manhole	60
Figure 4.15	Mislabeled Manhole	61
Figure 4.16	Indeterminate Orange Marking	62
Figure 4.17	Duct Marking Tag	65
Figure 4.18	Electrical Vault Tag	66
Figure 4.19	Qwest Manhole Tag	67

List of Tables

Table 4.1 High-Level net.Tagger Statistics 46

Table 4.2 Database Entry Format 52

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ACRA Application Crash Reports for Android

APWA American Public Works Association

API Application Programming Interface

AS Autonomous System

AWS Amazon Web Service

BOC Broadband Opportunity Council

CAIDA Center for Applied Internet Data Analysis

CBG Constraint Based Geolocation

CMAND Center for Measurement and Analysis of Network Data

DNS Domain Name System

DoS Denial of Service

DRoP DNS-Based Router Positioning

DWDM Dense Wavelength-Division Multiplexing

FCC Federal Communications Commission

FCC Federal Communications Commission

FDM Frequency-Division Multiplexing

FOC Fiber Optic Cable

FQDN Fully Qualified Domain Name

GIS Geographical Information System

GPS Geographical Positioning System

HIT	Human Intelligence Task
IP	Internet Protocol
ISP	Internet Service Provider
IXP	Internet Exchange Point
JSON	Javascript Object Notation
KML	Keyhole Markup Language
LAMP	Linux Apache MySQL PHP
NANOG	North American Network Operators Group
NGO	Non-Government Organization
NPS	Naval Postgraduate School
ORDBMS	Object-Relational Database Management System
OCR	Optical Character Recognition
OSM	OpenStreetMaps
PII	Personally Identifying Information
POP	Point of Presence
QoS	Quality of Service
ROW	Right-Of-Way
RTT	Round Trip Time
TBG	Topology Based Geolocation
TOS	Terms of Service
UI	User Interface

VPS Virtual Private Server

WDM Wavelength-Division Multiplexing

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

Few research projects with a title containing the word “Crowdsourcing” can occur without the support of many people, and this one was no exception. First and foremost, I would like to thank my advisors Rob Beverly and Justin Rohrer for their insights, persistence, and patience throughout this project. App development represents a new avenue in CMAND’s research strategies, and we quickly became aware of the complexities accompanying it. I am grateful to both of them for their support and involvement as we explored unfamiliar territory, creating what will hopefully become a valuable toolset for the future.

Many thanks to Professor Steve Bauer of MIT, who stands out as one of our earliest testers, a valuable source of feedback, and frequent tagger. I would also like to thank Jim Stewart, a family friend and experienced developer whose design insights brought the app interface from a barebones prototype to a usable product.

To all of our initial users willing to volunteer their time and personal phones for testing, I could not have finished this project without you. Special thanks to my family, who in addition to providing support and encouragement, became extremely well versed in the manhole covers and road markings of my hometown on this project’s behalf. Several of them have complained to me that they can no longer walk down the street without reflexively “looking for the Internet,” and I can only hope their newly acquired skills come in handy one day to make up for it.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

Physical network topology mapping represents a counterpart to mapping large-scale networks at more abstract levels. Many research groups have expended substantial efforts to map networks on the Internet Protocol (IP) level or higher. These efforts have resulted in a rich collection of data and tools useful for understanding the Internet's virtual structure. However, the underlying physical infrastructure of cables and the equipment they connect such as routers, data centers and Internet Service Provider (ISP) Point of Presences (POPs) is not as well understood on a fine-grained level.

1.1 Problem

It may appear contradictory that current research is better adapted to model the fluctuating virtual interconnections of the Internet instead of the static hardware that carries its traffic, but this is precisely the case for several reasons. Primarily, difficulties in mapping arise because the physical topology of a network need not match its virtual configuration. For economical, performance, and security reasons, trying to configure a network to match its physical makeup would be ill-advised even if possible. Traditional network mapping tools thus cannot be used for physical analysis without introducing substantial sources of error.

An additional hindrance to the availability of static hardware information involves the complex relationships between ISPs, public utility managers, and government regulators that leaves researchers without a centralized source of information. Large swaths of the physical Internet are installed, managed, and regulated by different parties that have little business incentive to communicate beyond their sphere of influence. Much of the information that would be beneficial to researchers is considered proprietary and not released by its corporate owners. Certain vendors compile and offer limited maps using ISP data, but this information is usually sold instead of made publicly available. Also, data pinpointing static hardware locations is based on what its owner claims is correct, usually leaving it physically unverified [1]. A final obstacle to advances in physical network mapping centers around the current publicly available data repositories that focus almost entirely on core

Internet backbone infrastructure [2]. A quality, publicly available, and consolidated source of low-level infrastructure does not exist at this time.

The prevalence of these challenges to network topology mapping has resulted in the rate of large-scale network expansion largely outpacing the ability of researchers to keep it physically catalogued. A strong argument can be made for the timely amelioration of these challenges, because understanding the composition and connections of the Internet not only provides valuable theoretical data to computer scientists, but is vital for the development of resiliency. Internet services play a central role in the commercial, government, and military sectors, and failures in reliability or performance have potentially serious consequences. Although Internet resiliency impacts both the national economy and security, it is not achievable without knowledge of the basic structure of the networks themselves. Because Internet traffic is usually consolidated in transit through several key points on the Internet's "backbone," a failure at any of these points could prove catastrophic. Comprehending the structure of the Internet gives both the government and industry the ability to diagnose weak points and build in redundancy where needed.

Critics of attempts to publicly map key network infrastructure contend these efforts serve as intelligence that attackers can use to plot operations. Their solution has been to either discourage extensive mapping or secure the results from public release. While a "security through obscurity" approach aligns with conventional military thought, the larger civilian security community sees this as a flawed approach. Their counterpoint normally states that true security lies in finding and fixing flaws instead of hiding them in hopes that they will not be discovered. The magnitude of this research problem is so great that multiple approaches from different research teams building on and collaborating with each other is necessary to yield results. Such efforts cannot exist without open exchange and publication of results.

Critics also need to be reminded that threats to the infrastructure can come not only from intentional sources such as terrorist attacks, but also from accidents or natural disasters. Undersea Fiber Optic Cables (FOCs) are frequently severed by boat anchors or other sources. In 2008, three cables were severed within days of each other in the Mediterranean and Middle East, reducing traffic capacity in some areas by up to 70% [3]. Natural disasters such as hurricanes or earthquakes can cause similar damage on land. Because these vulnerabilities are not predicated on human knowledge, not mapping or securing knowledge of them provides no benefits. They present risks similar to intentional attack or sabotage, with the best means of remediation being awareness of network structure so one may analyse for vulnerabilities in order to correct them.

1.2 Research Question

This thesis seeks to investigate several questions:

- What type and quantity of data must an app transmit to produce a useful data point? Given the constraints of an app transmission given available sensor data, privacy concerns, and bandwidth constraints, how can net.Tagger optimize a user's submission to gain enough information to reliably determine what exists at and what can be extrapolated from a given geographical position?

- What is the optimal User Interface (UI) to reduce erroneous submissions and provide user feedback? Within the realm of the user's experience, any interactions must produce accurate data and prevent the user from continuing to submit data out of boredom or frustration. Since the average user will not be able to identify telecommunications infrastructure indicators without training, the app must provide basic instructions on what to look for. Furthermore, crowdsourcing relies on the enthusiasm of its users to continue submitting based on whatever incentive they receive from participating. net.Tagger does not pay its participants, however initiatives such as the OpenStreetMaps (OSM) foundation have received open source mapping submissions from hundreds of thousands of unpaid volunteers without offering compensation. net.Tagger must be able to provide appropriate nontangible incentives or feedback to encourage participation and repeated submissions.

- How feasible is extrapolation from submissions to mapping inferences? net.Tagger works by identifying nodes based on user observations, but creating a map requires some means of correctly connecting the nodes. Based on initial data collection, how difficult is it to accurately generate map inferences?

1.3 Contribution

In addition to investigating the aforementioned research questions, the main contribution of this thesis project is creation of a working app/backend. Analysis of topics such as usability, data requirements, and findings analysis are explored, however this project serves primarily as the inception of net.Tagger, with the intent that future student researchers will further develop the initiative into a mature entity providing a previously unattempted approach to a major outstanding research area.

1.4 Thesis Organization

Chapter 2 provides an overview of existing physical mapping techniques, the crowdsourced mapping community, and telecommunications infrastructure types relevant to this project.

Chapter 3 lays out the framework of net.Tagger's different components as well as design choices and the actual project development.

Chapter 4 describes the testing methods used to evaluate net.Tagger and results from initial field testing and deployment.

Chapter 5 evaluates the conclusions of the project. Answers to the research questions are explored, looking at preliminary conclusions about applying crowdsourced mapping to network topologies. Given this project's scope as the foundation of a larger, ongoing initiative, future projects are described as well as a vision for an eventual large scale deployment of net.Tagger.

CHAPTER 2:
Background

2.1 Introduction

This chapter provides a brief survey of physical network topology mapping topics as they apply to this thesis. The structure of the Internet at a physical level is briefly described, with an emphasis on long-haul FOC conduits and the "Internet backbone." A number of policy-based decisions made within recent years are also explored as driving forces shaping the expansion of large-scale networks. These include Dig-Once laws, federal broadband expansion initiatives, and Right-Of-Way (ROW) lawsuits. Justification is given for the necessity of historical approaches to physical topology, including measurement-based strategies such as Constraint Based Geolocation (CBG) and DNS-Based Router Positioning (DRoP) as well as compilation-based approaches such as the Internet Topology Zoo.

2.2 Physical Internet Design

2.2.1 Organization

From a high-level perspective, the Internet can be studied and modeled at several levels [1]. The highest level is modelled in terms of organizations, which we define as entities under self control that are not subservient to other organizations. Based on structure and policy, each organization manages one or more IP prefixes known as Autonomous Systems (ASs). An AS is defined by RFC1930 [4] as

> a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy.

Because organizations often wish to divide their network assets into subsections to accommodate complex structures and routing policies, a complex organization will own and operate several ASs. Organizations do not just include ISPs, but can also be government and educational institutions, corporate enterprises, and content providers. At the AS level, these provider-level networks peer with each other at Internet Exchange Points (IXPs) and private points based on policy agreements [5]. The AS level is responsible for much of the truth behind the common networking idiom that "traffic does not follow the shortest path between two points, but the cheapest." At the POP level, an ISP aggregates routers and modems in a physical location (the POP itself) that provide a means for a local network of consumers to connect to the larger Internet backbone. The IP level consists of individually addressable end-hosts, aggregated subnets, and the router-level connectivity that joins the two. The IP level perspective of large-scale networks is frequently referred to as the "logical" layer, i.e., the organization and interconnections of individual network hosts depends upon the network's logical configuration instead of their physical location.

Finally, the physical layer consists largely of cables (fiber-optic or copper) and link-layer switching infrastructure. The physical layer can take other forms as well through mediums such as satellite Internet, however the core global Internet infrastructure utilizes FOC.

2.2.2 Long-Haul Geography

Because the logical topology of a network can be configured independently of its physical make-up, providers usually employ cost-saving measures to consolidate and share infrastructure. The "Internet backbone" is mostly comprised of FOC long-haul conduits, a term that is not precisely defined but can be generally described. One project [6] defined a long-haul conduit within the scope of their research as one either spanning at least 30 miles, connecting population centers of at least 100,000 people, or housing the cables of at least two providers. They define them more informally as "a 'tube' or trench specifically built to house the fiber of potentially multiple providers."

Long-haul conduits are frequently (but not unconditionally) placed adjacent to existing transportation infrastructure such as highways and railways. While expanding to meet growing consumer demand, long-haul networks can experience legal and logistical difficulties similar to other large-scale distribution networks such as railroads, power transmission lines, and petroleum pipelines. The mechanism that traditional utility networks utilize in many situations is the ROW, an easement between a landowner and a service provider seeking usage rights but not ownership of a section of private property. ROWs are characterized by binding legal contracts between the property holder and service provider that can be overseen by state commerce departments in order to ensure due process and equity, even in cases involving consensual agreements instead of eminent domain. However, lawsuits by property owners against ISPs show cases where ROWs were not observed in cases of long-haul FOCs laid alongside infrastructure such as rail lines. In 2013, Sprint Communications Co and WilTel Communications were ordered to pay $770,000 to 1,888 Connecticut property owners after the telecommunications providers negotiated with railroad companies to lay FOC along existing ROWs instead of negotiating with the property owners for a new easement [7]. Because the ROWs contracts only granted permission for the railroads to lay and operate tracks, the railroads were not authorized to grant Sprint and WilTel permission to lay cables. Similar suits have been filed around the country, with the Connecticut case the 35th statewide deal receiving final approval. Although Sprint has been utilizing this practice since the 1980s [8], the legal precedent now set by these cases could complicate placing FOC alongside transportation infrastructure in the future because telecommunications providers will have to obtain separate easements from landholders.

2.2.3 Traffic Consolidation

Studies of long-haul conduits frequently determine that conduit sharing between ISPs is a default practice. One study [6] "observed that 89.67%, 63.28%, and 53.50% of the conduits are shared by at least two, three, and four major ISPs, respectively." The same study found even more extreme examples, such as the conduit between Portland, OR and Seattle, WA that housed traffic from 31 separate ISPs. Traffic switching nodes also represent a point of traffic consolidation.

Traffic consolidation also takes place on the individual conduit level via several mechanisms. A single FOC cable contains many individual fibers, each capable of carrying traffic independent of the others. Due to the high cost of installing new cables, providers can simultaneously place more traffic on a single fiber through Wavelength-Division Multiplexing (WDM). WDM is analogous to Frequency-Division Multiplexing (FDM) due to the inverse proportionality of wavelength and frequency in electromagnetic radiation, however by convention WDM is normally used in reference to infrared frequency signals in optical media such as FOC, while FDM is used for radio frequency signals. By modulating separate data channels onto different carrier wavelength signals for transmission, WDM permits an FOC operator to send multiple messages simultaneously over the same fiber. Upon reaching their destination, the signals are separated via bandpass filtering and their messages extracted. Dense Wavelength-Division Multiplexing (DWDM), a subset of WDM, theoretically permits placing up to 100 10GB/s channels over optical media [9]. With each channel able to carry traffic from different senders running different networking protocols, WDM can consolidate substantial portions of traffic into the same physical conduits.

Another mechanism to move more traffic through the same physical location is "dark fiber." Because the high cost of installing FOC primarily lies in excavation, companies will frequently install more than necessary in an given conduit with the knowledge that a certain percentage of fibers will go unused for a time. Business transactions such as mergers and acquisitions among telecommunications companies can also leave providers with extra FOC running through the same conduit as live cables. These are commonly referred to as dark fiber, and can be leased to customers who desire a greater degree of control over their networks. Where WDM technologies can offer increase capabilities as a service, dark fiber operates as a physical asset. Leasing dark fiber gives a customer permission to operate these unused fibers as their own, with a wide degree of freedom in customizing their configuration.

2.2.4 Federal Initiatives

To encourage expansion and competition between broadband providers, President Obama signed Executive Order 13616 [10]: "Accelerating Broadband Infrastructure Deployment." The executive order provides funding and direction for government agencies to coordinate in order to streamline regulatory processes and reduce barriers experienced by broadband providers seeking to expand. The Executive Order covered a variety of areas, most notably initiatives known as "Dig-Once" practices [11]. When new broadband infrastructure (usually FOC) is laid underground in urban areas, up to 90% of installation costs are associated with the actual road excavation. This can create prohibitive expenses for ISPs seeking to expand into new areas, and also prevent new ISPs from entering markets in areas already covered by a single provider, depriving consumers of beneficial competition.

Dig-Once initiatives preemptively lay FOC conduits at the same time that new transportation infrastructure such as roads are put in. This permits ISPs to expand by running cables through existing conduits, avoiding the high expense of excavating from scratch. Proposals such as HR3805: The Broadband Conduit Deployment Act of 2015 [12] would mandate FOC conduits on federally-funded highway construction projects if the area in question is predicted to require broadband infrastructure within the next 15 years [13]. Although HR3805 has not been passed at this time, efforts initiated by EO 13616 are actively developing Dig-Once practices through other channels. As Dig-Once laws are more widely adopted, a side-effect will be further consolidation of traffic from multiple providers into the same channels.

In addition to Dig-Once practices, the Broadband Opportunity Council (BOC) established by EO 13616 made other recommendations that will shape the future growth of long-haul networks. The BOC's official report [14] pursuant to EO 13616 laid out several objectives, including:

- Make Federal lands and assets available for conduits.
- Standardize permitting and regulation, shifting it to the federal level to reduce burdens on local government and provide uniformity across state, local, and tribal boundaries.
- Emphasize broadband as an eligible and desirable funding target for community and regional infrastructure development projects.
- Collaborate with the private sector to reduce barriers to market entry and incumbent expansion for broadband providers.

Because federal efforts related to EO 13616 are still in their preliminary stages as of early 2016, most details regarding how government and commercial industry plan to implement and manage Dig-Once and related policies are not yet resolved. Timelines laid out by the BOC aim to resolve most details and begin implementing practices by the end of 2016. Regardless of their eventual form, federal efforts in this domain will only serve to increase the complexity of the national networking landscape, accelerating the need for improved understanding of both long-haul and lower level topologies.

2.2.5 Resiliency

The driving force for improved understanding of physical networks from a national security perspective centers around resiliency. With the increased dependency of vital services such as the financial, medical, energy, and transportation industries on network connectivity, disruptions have potentially disastrous ramifications. Over a sufficiently large period of time, a certain number of localized disruptions from man made or natural sources is inevitable. This forces government overseers and commercial providers to avoid working toward a perfect design in favor of one that can sustain damage and dynamically adapt to minimize downtime.

While traffic consolidation is an effective business strategy for scaling up network capabilities while maximizing profit, it comes at a price. When network traffic is constrained to a limited number of physical locations, infrastructure disruptions can produce greater outages than a more decentralized topology. During research for his book on the physical Internet, author Andrew Blum [15] visited a number of these locations, remarking at one that:

> This [room] was the main access point for Milwaukee's municipal data network, connecting libraries, schools, and government offices. Without it, thousands of civil servants would bang their computer mice against the desk in frustration. All this talk about Homeland Security, but look what someone could do in here with a chainsaw.

Damage to vital network infrastructure does not just come from malicious actors. In 2001, a CSX freight train derailed in Baltimore's Howard Street tunnel [16], causing a massive fire that burned for hours despite extensive efforts by emergency response personnel. In addition to causing property damage, the crash and subsequent fire severed a FOC conduit carrying Internet traffic from several providers as well as a large telephone FOC line. Although Internet access was largely unaffected in many Washington, DC areas, traffic between DC and west coast locations such as San Diego slowed by up to a factor of 10 in some locations. In order to restore redundancy, a team of telecommunications workers and city officials had to excavate the street in four locations to clear blockages and route 24,000 feet of FOC through manhole accessible conduits over 36 hours.

Natural disasters also pose a threat to networks that lack resiliency and redundancy. A Federal Communications Commission (FCC) independent review panel [17] of Hurricane Katrina's effects on communications networks identified line cuts and a lack of redundant pathways as two causative factors in the substantial outages accompanying the storm. One example from their findings was a long-haul FOC conduit with a tandem switch inside New Orleans and paths out of the city to the east and west. After the eastern route was cut by a barge blown ashore, the western route was cut first by falling trees, and later by construction crews removing debris from a highway ROW. Damage to a small number of switches in New Orleans impacted traffic both inside the city and on conduits linking regions of the country. Accidental fiber line cuts by clean-up and response teams were so prevalent that BellSouth reported major routes cut in multiple places, and Cox Communications estimated that 11 days after the storm it had suffered more network outages due to human damage than the storm itself.

2.3 Physical Topology Mapping History

While many details remain unanswered, physical topology mapping research is not without its past efforts. Since the early 2000s, many research groups and private companies have attempted to make progress, with substantial but still limited successes. Most research initiatives in this area fall into one of two categories. Measurement-based projects attempt to directly calculate results, normally by sending probes to certain destinations and timing the responses while trying to compensate for errors induced by propagation, queuing, and virtualization. Compilation-based projects rely on seeking out preexisting data from different sources that independently offer little insight, but by gathering them together and analyzing them, yield new results.

Many research projects addressing physical topology mapping are not fully applicable to the problems projects such as net.Tagger seek to address. Most work focuses on IP geolocation, which seeks to identify the rough geographical position of individual IP addresses or IP subnets. IP geolocation has many commercial applications including targeted web advertisements, fraud protection, and determining the applicability of interstate or international laws [18]. However, conventional IP geolocation suffers from two shortcomings regarding physical topology mapping. First, the level of accuracy is normally too low.

Even commercial geolocation services are usually limited to placing IP addresses within a given zip code or greater, which is insufficient for constructing fine-grained maps [19]. Second, much of the desirable infrastructure targeted by researchers for mapping exists below the IP layer [6]. The physical infrastructure sought by this project and other similar ones cannot be completed by simply identifying the probable locations of router or higher level architecture.

2.3.1 Measurement Based

One approach to network topology mapping that has been studied and expanded upon for years uses a variety of probes and timing measurements to roughly geolocate individual IP addresses and small subnets. These methods employ a number of "vantage points," consisting of servers (such as PlanetLab nodes) at precisely recorded coordinates to send probes to target hosts. The propagation delay of FOCs is relatively fixed at 2/3c, which increases to 4/9c when factoring in transmission, processing, and queuing delays.

The most basic implementation of timing-based geolocation was used by early implementations such as GeoPing, which made the observation that if the Round Trip Time (RTT) between two known hosts was similar to the RTT of one of the known hosts and an unknown target host, there was a tendency for the two to be geographically clustered [20]. These techniques relied on a large number of assumptions that their authors readily admitted, but they represented some of the first efforts into IP Geolocation in the early 2000's. Accuracy with this basic implementation was limited, with GeoPing requiring 7–9 probe sources to achieve an accuracy in the 100's of km.

Fortunately, the past 10–15 years has seen a number of improvements. One of the most important was the publication of CBG in 2004 [21]. Unlike earlier methods that could only produce a discrete number of possible positions equal to the number of reference hosts, CBG is capable of using multilateration to place a target host in a probable region that may not include any of the reference hosts. Despite representing a substantial improvement with room for growth, CBG is effectively limited to a median accuracy of 228 KM. Combining CBG with high-level knowledge of ISP topology gained through other sources resulted in the creation of Topology Based Geolocation (TBG), with an improved median accuracy of 67 km [22]. Further augmentation with knowledge of router locations and demographics data permits tools such as the Octant framework to achieve a median accuracy of 35.2 km [23]. While research continues to improve IP geolocation to the point that it may be used for limited topology discovery [24], it still suffers from the shortcoming of targeting too high a level of the Internet too inaccurately to produce the fine-grained, low-level maps that would prove most beneficial to researchers.

Another IP-level geolocation method that augments timing-based approaches is DRoP. DRoP takes advantage of common naming trends within the Domain Name System (DNS) protocol, which maps human readable domain names to network addresses. Although no official standard naming convention exists for DNS, the hostnames of router interfaces can include descriptive keywords selected by the infrastructure's owner to assist the organization and administration of their assets. Frequently, at least some of this information will include geographical hints about a location holding the physical infrastructure pointed to by a DNS entry. Most are fine-grained to the city level. Common examples include

- IATA/ICAO codes identifying the largest airport in a city.
- CLLI position codes carrying varying levels of geographic resolution, normally truncated to city/state for domain names.
- UN/LOCODE, identifying specific locations of locations relevant to the shipping and manufacturing industry. Developed for European commerce.
- City names or abbreviations.

However, utilizing hostname hints for geolocation is far from straightforward. Many hostnames contain multiple pieces of information that could be interpreted as data with no way to determine if the hostname owner chose any to describe the item's location. An example given by Center for Applied Internet Data Analysis (CAIDA) is the hostname *ccr.par01.atlas.cogentco.com*, which potentially contains a Connecticut airport code (ccr), a reference to Paris (indeterminate country), or a possible reference to Salas Atlas in Spain. All hints point to different locations, and the hostname alone does not give sufficient background on the holder's naming convention to say if any is correct. Despite these ambiguities, DRoP hostname data can still provide useful insights. One approach is to group hints based on their domain level (inferring possible similarities in naming schemes) and then check possible guesses against timing-based measurements to enact constraints based on latency data. Combining timing measurements with DNS hostname has the potential to provide accuracy down to the level provided by the hostname hint (usually the city containing the interface), however DRoP is ineffective if an interface lacks a Fully Qualified Domain Name (FQDN) or if nothing in the hostname matches a known hint. Previous work places the number of router interfaces that cannot be classified with DRoP at approximately 45%.

Combining measurement and compilation methods can infer additional relationships beyond geolocating individual network nodes. Giotsas et al. demonstrate a method for mapping AS peering connections to facilities that makes use of several geolocation methods. They begin by manually compiling a database of facilities such as IXPs and the networks present at them. This information can be gathered primarily through self-reported data published by the facilities to advertise the networks they support to peer with.

2.3.2 Compilation Based

Another approach to physical topology mapping relies on gathering data from existing sources. Even though central repositories of topology data are not readily available, focused subsets do exist. One source of data are the maps published by Tier-1 ISPs themselves. ISPs frequently distribute rough maps of their central FOC graphs as commercial promotions to demonstrate to potential clients the scope of their coverage. These maps provide a general survey of their routes, but they frequently omit router-level detail, as ISPs consider such information proprietary. Researchers who utilize them also observe that these maps are sometimes optimistic, over-simplified, or out of date.

Tier-1 ISP maps are still of use to researchers as a starting point. Some projects have successfully started with ISP maps and fleshed out smaller details through clever use of other data sources [6]. A 2015 project [6] combined ISP maps with geocodings from the Internet Atlas Project [2] to create a base map. The researchers then exhaustively gathered public domain information such as government/municipality records, commercial entity documentation, utility ROW, environmental impact statements, and fiber sharing arrangements from states' Departments of Transportation (DOTs). Through extrapolation and cross-correlation, the team was able to produce a number of conclusions about the state of long-haul FOC infrastructure and the sharing agreements implemented by ISPs on the physical level. Provided the underlying documentation and extrapolation assumptions are correct, mapping efforts like these provide a valuable counterpart to the error-prone measurement based techniques. However, the quantity and variety of documentation used for these projects makes validating their accuracy infeasible. They also tend to focus on larger Internet backbone infrastructure because their methods and the documentation they rely on do not accurately scale down to more fine-grained levels.

Another area of compilation-based network mapping with a much more established history is that of submarine communications cables. A successor to submarine telegraph and telephone cables, modern submarine FOC cables carry the majority of transcontinental Internet traffic. Because of their crucial role in connecting countries to the global Internet backbone, submarine cables are considered by many governments as vital national assets. However, submarine cables are frequently subject to damage due to natural phenomena such as ocean current and earthquakes or manmade sources such as anchors, trawling nets, or intentional sabotage. Their importance, vulnerability, and relatively low numbers make submarine cables a sought-after mapping target by telecommunications research firms who sell maps and data to a variety of customers. Various free sources exist such as TeleGeography's interactive online Submarine Cable Map [25]. However, most free maps are deliberately designed with a low level of detail. TeleGeography's free product is "stylized to improve readability" and "does not reflect the physical cable location." Its cable landing stations are also "not precise coordinates" and "are meant to serve as a general guide." More descriptive maps and datasets are available from these sources but come with expensive subscription fees and licensing restrictions on use.

2.4 Crowdsourced Mapping

Much of the initial inspiration for net.Tagger came from the success of crowdsourced mapping projects, the most notable of which is the OSM project [26]. OSM is a worldwide initiative with its origins in Europe, officially supported but not managed by the OSM Foundation. Its goal is to provide a freely available, open source collection of GIS data. Often described as the "Wikipedia of Google Maps," OSM has over 2.4 million registered users [27] submitting data. OSM users gather data through different means and submit their findings to OSM using one of many available web, desktop, or mobile editor applications [28]. Most of the editors are created through community projects with OSM's publicly available editing Application Programming Interface (API) and provide experiences designed for subsets of the user base. Although many different options exist for users to interact with the OSM data set, the three most popular editors are iD, Potlatch2, and JOSM. iD and Potlatch2 are both browser based editors available directly from the main OSM website's planet map. They permit users to tag and edit as they interact with a map populated from the entire OSM dataset. Potlatch2 is an older editor that requires flash browser support, however it offers more features than iD and is still widely used. iD is javascript based and is designed for more novice users, with an emphasis on simplicity. JOSM is a standalone desktop application designed for experienced users, providing customizability through plugins and a broader feature set at the price of a steeper learning curve. JOSM allows users to input large data sets offline, automatically validate for common errors, and then push edits to the OSM dataset when finished. Although these three editors are the most common among the OSM community, many other open source editing applications exist that make use of OSM's editing API. OSM's editor documentation [28] currently lists seven editors apiece for android and IOS devices. The smartphone editors vary in capability and intent. Some are designed for other Geographical Information System (GIS) purposes and offer limited ability to push edits to OSM, while others are fully feature editors capable of submitting all types of OSM objects from field locations. After OSM received permission to overlay satellite images from sources such as Bing Maps over its existing tiles, users became able to visually identify and trace out features on these applications without needing to conduct field surveys.

Because OSM relies on the assumption that users will vet data before submitting, most of their data error come from inadvertant user mistakes or intentionally placed copyright easter eggs [29]. The official OSM wiki [30] addresses this issue by noting that even proprietary data sources have errors including intentional "copyright easter eggs." It also discusses the "wikipedia-style model" the project follows, where each user can add history/submission metadata to their profile's uploads. OSM claims that because most users are deliberate in their methods and non-malicious, the collection of correct data points is substantially larger than the few incorrect ones, and overlap between user submissions will quickly identify and correct small errors.

Formal analysis of OSM data shows that these claims are reasonably correct, with several caveats. One study [31] compared formal geographical survey data against data from OSM and Tele Atlas, a commercial GIS supplier to many projects including early version of Google Maps. Analysis found that both OSM and Tele Atlas deviated from the survey data with similar spacial deviations. However, OSM showed greater inaccuracies in rural areas, where the study deduced that there were fewer users than urban areas where the OSM error rate was comparatively lower. Another study [32] found that the majority of high quality OSM submissions came from a core group of "Expert" to "Professional" level users comprising only 3–4% of the OSM user base, with an accuracy approaching or at the level of commercial agencies. The lowest levels of participation and submission quality came from the approximately 74% of "Beginner" users. In addition to user-submitted findings, OSM utilizes imports from many other open GIS repositories [33] with the permission of the owner, providing a foundation of data from a multitude of sources, many of which were professionally gathered. The OSM dataset is used by private citizens, companies, and government agencies for web, desktop, and mobile applications. Proprietary GIS datasets potentially come with licensing fees, Terms of Service (TOS) agreements, and privacy policies that are incompatible with the fiscal resources or ideological viewpoints of application developers or their userbase. Because of its open source philosophy, OSM is free to use and under the Open Data Commons Open Database License, has a very liberal use policy that only requires attribution to the OSM project. By contrast, most Google Maps API developer tiers permit small-scale usage for free but begin charging an owner once registered applications using their API key exceed 25,000 queries per day. Although Google offers Quality of Service (QoS) guarantees and additional support with its higher priced tiers, many small open-sourced projects requiring a GIS dataset are minimally funded and utilize the expertise of its user base for technical support. For them, relying on proprietary systems is infeasible, and OSM data combined with free GIS software libraries allows them to develop at minimal cost. As a result, small independent developers have produced a plethora of OSM reliant applications from smartphone navigation apps to online search engines for National Park campsites. OSM is also utilized by government and Non-Government Organizations (NGOs) for crisis mapping. After the 2010 Haiti earthquake decimated large swaths of the country, rescue teams were hindered by the lack of accurate, up-to-date maps.

OSM volunteers began recording roads based on available Yahoo imagery. Other volunteer teams deployed to Haiti itself to begin mapping with OSM techniques. The end result was a highly detailed GIS resource that quickly became the default map for all NGOs in the area as well as other responding organizations such as the United Nations and the World Bank [34].

Crowdsourcing has also been applied to networking projects with success. The Portolan project [35], a collaboration between Italian research entities including the University of Pisa, is one such example. Portolan employs a distributed smartphone app framework similar to the one proposed by us for net.Tagger. It seeks to build maps of mobile device signal coverage and AS-level connections by collecting a combination of passive and active measurements from smartphone sensors. The Portolan app utilizes geolocation measurements from other onboard phone applications to minimize battery use, correlating them with time-synchronized measurements of phone signal strength [36]. The app also performs traceroutes to target locations after receiving periodic instructions from a central command and control server that also collects and stores data. Portolan's creators identified a streamlined and minimal user experience, low smartphone resource footprint, and providing users with access to a partial results dataset as their main design goals to encourage user participation [37]. They selected Android as their initial deployment platform, citing an overall ease of development and distribution that outweighed the difficulties in implementing networking algorithms such as Paris Traceroute. Preliminary analysis of Portolan research results showed consistency against a CAIDA traceroute dataset and even several cases where traceroutes from smartphones employing the app traversed routes in the opposite direction as the CAIDA traces, uncovering new router interfaces. Although Portolan is still in its infancy relative to its developers' eventual objectives, it demonstrates the utility of performing crowdsourced, smartphone-based network measurements.

2.5 Infrastructure Indicators

net.Tagger's basic approach to physical topology mapping relies on a user's ability to identify street-level indicators of telecommunications infrastructure. This presents two challenges: First, users may not have previous experience in spotting infrastructure, and second, most infrastructure is hidden from view and can be identified only through indirect indicators. Most indicators available to common observers signal the presence of FOC. Exceptions exist, but most sensitive equipment such as routers or server racks are secured on private property owned by ISPs. However, the connections between these entities often pass through public space, and must have some means for their owners to access them to perform maintenance. They also must be marked clearly enough that other contractors or utility providers do not inadvertently damage them during construction or operations. Publicly available information on telecommunications markings is limited, but a combination of public utilities publications and field research performed for this project has revealed the following targets of interest for net.Tagger.

2.5.1 Orange Markings

One of the most prevalent and reliable street-level indicators of telecommunications equipment relies on the public utility color-coded system. The system is maintained and promoted by the American Public Works Association, a non-profit professional organization including both public works agencies and private sector companies who work in the field. The American Public Works Association (APWA) Uniform Color Code [38], laid out in ANSI standard Z535.1: Safety Colors for Temporary Marking and Facility Identification (see Figure 2.1), is not absolutely binding but is followed by most agencies throughout the country for conformity reasons. The purpose of the APWA Uniform Color Code is to standardize the markings public utility agencies and companies use to identify and warn each other of the presence of underground infrastructure based on type.

Figure 2.1: Street Markings Color Code. Source: [39]

The most relevant color entry for net.Tagger's work is orange, specifically color shade PMS 144. Bright orange markings laid in paint or chalk on roads, sidewalks, or other public spaces in the United States are usually a sign that telecommunications equipment is present below ground. This can include phone lines, cable TV, or fiber-optic cables. The markings vary greatly in style depending on the project, but will frequently be drawn with lines or arrows indicating the direction of travel of the cables. Many have amplifying information including the ISP who owns the equipment and what their particular use is. Figure 2.2 and Figure 2.3 show examples.

Figure 2.2: Orange Marking

Figure 2.3: Orange Marking

Even though phone and cable TV lines are not of primary interest to the net.Tagger project, multiple types of cables are often run together to economize on space, thus any orange markings are a desired find. Even better are markings carrying the initials “FOC,” indicating fiber-optic cables. If a marking specifically states fiber-optic, there is a higher probability it carries network traffic instead of other services. Assigning this higher certainty to a find creates a more useful data point for later topology extrapolation.

One other street marking color of lesser significance to net.Tagger is white, indicating “proposed excavation.” Because white does not specify if the excavation is for telecommunications work or other purposes, white markings alone are useless for net.Tagger. However, the field research conducted for this project frequently found white markings that were covered over by orange, suggesting that excavation occurred and telecommunications equipment or cabling was installed. This can provide a potentially useful data point regarding the recency of the find. It is important to note that these criteria do not apply outside of the U.S., where different color codes are used. For example, in the UK, telecommunications equipment is identified with the color green, which in the U.S. indicates sewers and stormwater systems.

2.5.2 Duct Markings

Orange street markings come in a variety of shapes depending on their intended use. One subset of orange markings is of special significance because they indicate a duct carrying a bundle of telecommunications cables. Duct markings also have several forms they can take, but most consist of several parallel lines or parallel lines boxing in a diamond as shown in Figure 2.4 and Figure 2.5.

Figure 2.4: Duct Marking

Figure 2.5: Annotated Duct Marking

Frequently duct markings will be annotated with the width of the duct (such as “24 inch FOC duct”). The personnel laying down duct markings will usually string together several markings in a line, indicating the exact location of the communications channel. Duct markings have the benefit of identifying a greater than usual concentration of telecommunications infrastructure as well as exactly where it leads to, giving valuable information to prospective mappers.

2.5.3 Manhole Covers

Accompanying temporary paint or chalk markings are more permanent infrastructure indicators that serve as access points to equipment for maintenance personnel. The largest and most prominent examples are manhole covers. Although many manhole covers in an urban area provide sewer access, others are devoted to accessing telecommunications equipment. Unlike sewer accesses which are marked with “Sewer” or “S,” telecommunications manholes will bear the name of the provider who operates their underlying equipment. Most will also bear a unique, distinguishable honeycomb pattern visible in Figure 2.6 and Figure 2.7, but other categories of manhole covers (such as those used for accessing power equipment) might also have this pattern. In addition to the middle of streets, telecommunications manholes can be found on sidewalks and in the middle of traffic intersections next to sewer accesses.

Figure 2.6: Bell System

Figure 2.7: US West

Manhole covers do not provide as detailed information as other sources, but they still identify the presence of telecommunications infrastructure at a location. The operator name that they provide is also useful data, however the markings do not necessarily reflect the current owner if the original owning company was bought or sold.

2.5.4 Handholes

A less prominent, but often more descriptive maintenance access point, is the handhole. A smaller cousin to manholes, handholes are usually found on sidewalks and are much smaller, only providing enough room for a technician to reach inside instead of enter entirely. Similar to manholes, handholes might be used for different equipment such as power or water meters. Telecommunications handholes can be marked with the name of their equipment owner, but often bear descriptive names as well (Figure 2.8and Figure 2.9). Some are stamped with their specific purpose (“Broadband,” “Cable,” or even “Computer”).

Figure 2.8: Communication Handhole

Figure 2.9: Computer Handhole

Others are even larger, approaching the size of manhole covers and bearing additional information such as the ratings of the equipment they protect. Figure 2.10 and Figure 2.11 both demonstrate equipment ratings labels.

Figure 2.10: Fiber Optic 15/20K

Figure 2.11: SBC NewBasis 20K

Handholes provide similar information as manhole covers, with the occasional bonus of amplifying information.

2.5.5 Dig Warnings

The infrastructure indicator that most non-technical persons are familiar with are "Call Before You Dig" signs erected to warn landscapers, homeowners, and contractors about the presence of buried hazards such as gas lines. Telecommunication dig signs can frequently be found along roads and are usually small green or gray columns with an orange sign stating "Warning: Underground Cable. Dig Safely" and giving the name of the provider managing the cable. Figure 2.12 and Figure 2.13 show different dig warnings on similar columns.

Figure 2.12: Qwest Warning

Figure 2.13: Century Link Warning (Close-Up)

Although dig warnings might seem to provide a limited amount of information, they sometimes permit helpful data extrapolation. Because FOCs usually (but not always) follow roads, a string of dig warnings along the same section of main road labeled with the same provider name is a strong indicator of the direction the cable lies in.

2.5.6 Cell Towers

Some cell towers are easily identified by by signage placed on surrounding fencing that lists operator names and the tower's FCC identification number. Figure 2.14 shows a standard cell tower base with its accompanying labelling.

Figure 2.14: Cell Tower Markings

Others are deliberately concealed to blend in with local landscapes and features. In Figure 2.15, a cell tower has been disguised as a tree, although its distinctive base is still visible.

Figure 2.15: Hidden Cell Tower. Source: [40]

This practice allows providers to place infrastructure in close proximity to urban areas, however local residents sometimes file lawsuits over supposed health effects [41]. Online communities [42] exist devoted to cataloging examples of cell towers in a variety of disguises ranging from cacti to church steeples. While some are fully concealed, others are still surrounded with standard fencing and FCC markings that can be easily identified by a nearby observer. Even though the cell tower in Figure 2.15 is disguised as a tree, its distinctive base is still visible. Figure 2.14 shows a different tower that is not concealed, demonstrating the full range of labels that might appear. Cell towers are useful in mapping because they are frequently connected to sizable ground FOC lines. Searching the roads and trails surrounding a cell tower usually leads to discovery of other infrastructure indicators in the immediate vicinity. Cell towers represent a useful location to begin a fresh search for infrastructure and can be good jumping off points for further investigation.

2.5.7 Buildings

Buildings holding actual infrastructure equipment such as servers, routers, or data storage are difficult to identify because they are usually well-secured on private property and unmarked. In the event that following FOC trails leads to identifiable ISP properties, a very useful mapping association is made. net.Tagger allows users to submit building findings in the event that a possible building is identified due to the potential value of the find.

2.6 Android Platform Capabilities

The net.Tagger concept relies on a distributed network of smartphones that can individually collect and submit research data. We utilize Android for our initial development and release. In addition to comments and other data that users can enter manually, the platform provides the following capabilities.

2.6.1 Location Data

Android currently offers two location APIs. The first is the stock Android.Location API [43], which is still supported, but in the process of being phased out. Google recommends developers utilize the newer Google Play Location Services API [44], which requires registration with the Play Store but offers better performance, accuracy, and battery usage.

Either API can be interfaced with the Google Maps API, which requires additional registration but permits an app to directly display location overlays. Developers can configure "Location Listeners" at runtime that dictate how frequently and precisely the app performs location updates, trading accuracy for battery usage.

2.6.2 Sensors

So long as its underlying hardware supports all sensors, an Android smartphone app can collect raw data from many types of sensors [45]. Not all devices will contain all possible sensors, and some devices may contain multiple instances of the same sensor that have different levels of precision. The Android sensor management packages provide tools for an app to determine which sensors exist on a device, what capabilities those sensors have, and how to register and read from the sensors. Examples of Android sensors [45] include:

Motion Sensors

Gyroscopic, accelerometer, and rotational vector sensors that can measure rotation and translation in all three spatial dimensions.

Environmental Sensors

Barometers, thermometers, and photometers that can measure humidity, atmospheric pressure, temperature, and illumination.

Position Sensors

Orientation sensors and magnetometers that measure the physical position of a device.

2.6.3 Camera

Although the Android Camera API permits fine-grained control of any onboard cameras, it also provides built-in tools to use basic camera features with minimal effort. Android documentation recommends that developers determine the role that image collection plays in their project and utilize these pre-existing tools unless their app requires a custom camera configuration. The Camera API permits developers to integrate the stock camera UI that all users are familiar with into their apps, which reduces the possibility of user error or stability issues accidentally introduced by developers.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Implementation

3.1 Project Requirements

The core goal of the net.Tagger project is to obtain GIS data and descriptions of street-level network infrastructure indicators in sufficient quantity and detail to infer accurate insights about underlying network topology. net.Tagger will pursue this goal via a distributed crowdsourcing approach that is easy and fulfilling for the project's user base. Crowdsourcing will be implemented via a mobile app. For our purposes, we consider an app as a program running directly on a mobile device's operating system [46]. This is in contrast to software running on a dedicated computer or through a web browser. Core project requirements (in no particular order) are:

- The overall app experience should be as streamlined as possible to minimize user frustrations, reduce the app's learning curve, and increase the likelihood of a user's continued involvement in the project. Most users who seek to become involved will possess some networking knowledge, however their initial unfamiliarity with net.Tagger and the project's target data must be overcome to produce productive users. A straightforward user experience will lower barriers to entry and reduce opportunities for a user to execute the submission process incorrectly. Similar to OSM's crowdsourcing process, our project model contains a possibility that users will misinterpret findings or improperly perform submissions. A simply, streamlined user experience introduces fewer opportunities to perform an erroneous action. Overall, the app should be able to move a user from identifying a finding to submitting a data point in the fewest number of interactions (such as clicking or entering text) as possible.
- The app must send enough data on a tag submission to provide a useful data point. If the ultimate goal of net.Tagger is to infer meaningful and accurate network topology data, certain key pieces of information are necessary for each submission. At a minimum, a "tag" is a single transaction sending Geographical Positioning System (GPS)

coordinates, the GPS accuracy at time of submission, a timestamp, and the user's belief about the infrastructure's type and provider. The user must also be encouraged to submit images and any miscellaneous observations, providing extra resources for net.Tagger researchers to verify submission accuracy and make network inferences.

- The app must provide users with text or graphical feedback immediately after submitting a finding. The feedback will ensure that users see that their action completed, keeping them invested.
- The app experience must provide users with incentives to continue participating. A multifaceted approach should be employed to reach users with different motivations. These can include community prestige through an online leaderboard, small monetary rewards, or providing access to a portion of the dataset in exchange for participating. These incentives should be tailored to improve the quality of research data, such as providing additional rewards for validating existing tags from other users instead of just submitting original tags.
- The app must operate reliably, handling errors properly, and avoid crashes. Stability issues are likely to induce frustration in users, leading to reduced participation or quitting the project altogether.
- The app must balance user privacy, data security, and overall usability. The app should maintain a unique profile for each user used to identify and authenticate their data submissions, but limit required user information to that necessary for research purposes. No information should be collected without the user's knowledge and consent.
- Data submitted by users must be protected during submission ("in transit") and in storage ("at rest"). Data must be secured in transit against an adversary capable of intercepting cellular signals or sniffing network traffic. Data should be stored on servers we control, and in a manner that is resistant against web and database attacks (such as SQL injection). No services or databases should be not be exposed beyond what is necessary for approved client/server operations and additional access must require administrator credentials. secure.
- Data should be logically ordered in order to facilitate indexing, retrieval, and interfacing with standard GIS tools such as the OSM software stack. This does not affect the data collection process, but is required for the eventual data analysis that is the

core goal of net.Tagger. Because the eventual dataset will be very large, it must be stored in a format that can be efficiently queried based on parameters and constraints via native PostGIS functionality, scripts, and GIS software.

- At a minimum, users should be able to view their own tag history directly from the app. Ideally, users should also be able to view the entire set of tags both from the app and online if resources permit.

net.Tagger's design requirements were chosen to support two approaches to user data collection. As the OSM project demonstrates [32], the most accurate and complete data will likely be submitted by a small, core section of users. This group will likely possess greater than average technical knowledge and a willingness to devote blocks of time and effort specifically to collecting data. These users will be interested in submitting findings that are not only accurate, but also as complete and informative as possible. If the app offers extra functionality, they are likely to learn and use it properly. They will also be concerned with their search coverage, canvassing as large an area as possible without missing or repeating sections.

Similar statistics on OSM users shows that a larger proportion of users will contribute less frequently and with a higher chance of submitting incorrect or incomplete data. These users will benefit from a simple experience that requires a minimal amount of time and number of interactions to submit tags. Their submissions are likely to be made while conducting other activities, making convenience and usability key to their continued participation. They do not require complex features, as they are less likely to take the time to learn and use them regularly.

Most users will not fall explicitly into one of these two groups, but will use a combination of both methods depending on their lifestyle. A user might perform detailed, structured data collection for several hours on a weekend but also submit findings as they come across them during weekday activities. To capitalize on its user base, the app must cater to both methods. The UI and user experience must be streamlined enough for quick and intuitive submissions, while still allowing users to track their past submissions and provide additional details when they have the time and interest to do so.

3.2 App Design

3.2.1 Initial App Design

During its original development, the net.Tagger app focused on function over form. As the project evolved and received input from reviewers, several UI necessities became apparent. Initial iterations of net.Tagger were structured as follows:

- The user began on a "main screen" (Figure 3.1) that linked to pages such as profile data, data submission, instructions/examples, and a display of past submissions.
- After setting up a profile and viewing the training pages, the user spent most time on the data submission page (Figure 3.2) to submit findings.
- To receive any feedback beyond a "Data Submitted" message, the user needed to take several extra steps that brought them out of the submission cycle.

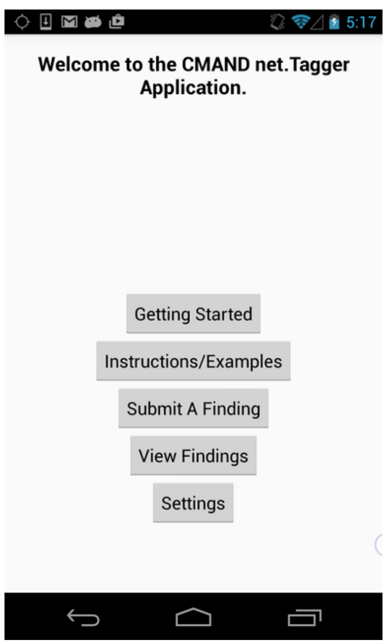

Figure 3.1: Initial Main Screen

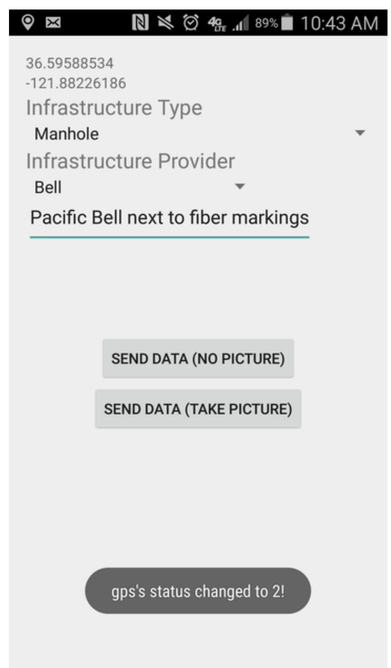

Figure 3.2: Initial Submit Screen

The layout was not conducive to a positive user experience and was likely to foster disinterest and frustration. The barebones prototype was adequate for initial development, but did not meet all design requirements.

3.2.2 Refined App Design

- The main page (Figure 3.3) that the user "lives in" was changed to include the submissions map. This ensured that the user constantly sees their previous tags and is immediately shown the result of a tag submission as a new map marker. The user can also watch their position marker move around the map filling in blank spaces with fresh findings. This provides constant feedback without moving to a fresh app screen.
- Tasks such as submitting data, modifying profiles, and viewing infrastructure indicator examples are moved to pop-up activities that display off of the main app screen (Figure 3.4). The user does not have to click through multiple screens to accomplish basic tasks, reducing time away from the main screen. All interactions take place from a single, central screen.

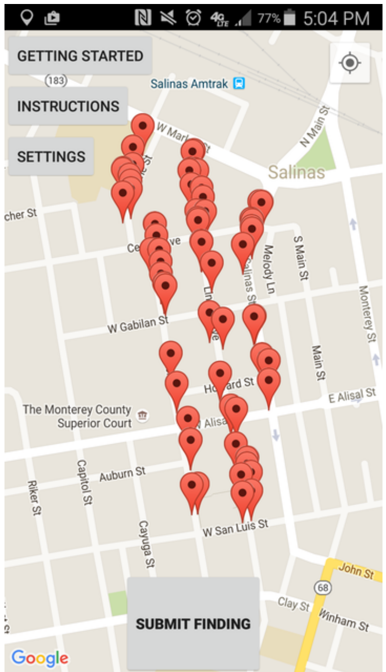

Figure 3.3: Refined Main Screen

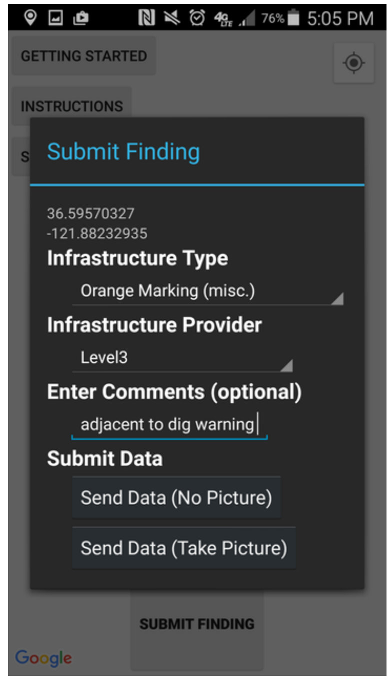

Figure 3.4: Refined Submit Screen

Figure 3.3 shows a user's main screen after two hours of tagging in downtown Salinas, CA.

3.2.3 Platform Selection

Because crowdsourcing depends on reaching the largest possible user base, net.Tagger would ideally be developed for multiple smartphone architectures. However, confining the project to a single architecture for initial research phases facilitates testing non-app components without the substantial workload brought on by deploying on different platforms. A mature project can only be created through continual deployment and testing that reveals issues needing resolution. This necessitates choosing a single smartphone architecture for initial app development before porting to others. Early testing before wide scale deployment does not rely on reaching a broad user base, placing a premium on platform development ease instead of overall market share. After considering available options, Android and IOS emerged as the most viable architectures for an initial net.Tagger app. Android's documentation, developer community, open source philosophy, and distribution system made it an ideal development platform. Although either option would have worked well, easy integration with tools such as the Google Maps API and the Google Play Store reduced many project requirements to previously solved problems. We leave expansion to IOS as future work.

3.2.4 User Interface

The most important iteration in the evolution of the UI was bringing an emphasis on feedback to the forefront of the user experience. Early versions were successful in gathering data during local field tests, however the testing was carried out by project members with external motivation to continue submitting data. With this configuration, a normal user without any explicit ties to the project would be expected to expend time walking around urban areas entering data about their finds without receiving immediate feedback beyond a "Data Submitted" app message. Most users would quickly grow disillusioned with this configuration, feeling they were performing unpaid labor with little incentive to continue. A successful crowdsourcing project depends upon users feeling invested in a common goal, and the early app UI did not accomplish this.

Several different solutions to the user feedback problem were evaluated for feasibility versus payoff. For example, an approach requiring minimal effort would be to run scripts on the net.Tagger Virtual Private Server (VPS) to let users download a Keyhole Markup Language (KML) record of their submissions to view in Google Earth via a tablet or PC. This basic solution permits the user to view submissions, but only after returning from gathering data and completing several steps. We posit that a dedicated group of users might be willing to perform these extra tasks to view the results of their efforts, but this might discourage more casual users. It also violates our design requirements that emphasize a streamlined process with immediate, automated user feedback.

Another prototyped solution kept a KML file on the user's phone to record submissions locally in addition to sending them to the net.Tagger backend server. After making a series of captures from the "Data Submit" page, the user had the ability to return to the main page and select a "View Submissions" option. This would launch Android's Google Earth app (assuming the user had it installed on their phone) and load the local KML file, displaying the user's submission history as a series of map markers overlaid on a global map. This approach provided the user with instant smartphone feedback identical to the previous option. The user no longer needed to download a file and could view their map in between submissions, even while gathering data. However, this design had several drawbacks.

Due to the design of the Android OS, opening Google Earth and populating it with net.Tagger data was a trivial task. However, if the user was already running Google Earth in the background when they tried to view submissions in net.Tagger, no new data would be loaded. As a stopgap, the app displayed a message to the user reminding them to close instances of Google Earth before viewing tag submissions. Counting on a user to follow extra task direction for a basic feature to work properly is inadvisable and risks frustrating users. A good UI design should present immediate feedback within one to two seconds every time a user performs a task, particularly a data submission. Although this design was an improvement over the initial layout, it still required a user to submit tags from one screen, navigate to the main page, leave the app to check the Task Manager, return to the app, and select “View Submissions,” opening up an entirely separate app (Google Earth) to finally display findings.

The final UI layout came about after gathering feedback from test users, some of whom had prior app development experience. The most important design decision was changing the workflow to shift the submission map from a secondary feature to the app’s primary focus. All previous iterations of the app required the user to begin at a main page and navigate between separate pages to submit and view findings. A streamlined design put the submission map as the main page, with the user navigation to other pages through the map screen. This was made possible through integration with the Google Maps API. By utilizing an Android MapView as the background of the main page, the user’s default view is now a map overlay that shows their position and instantly populates itself with markers after each submission. An eventual development goal is to populate each user’s in-app map with a rough representation of the entire net.Tagger dataset, showing them all covered and uncovered regions. However, implementing this feature in the app’s initial release was infeasible due to time constraints so a local map of the individual user’s finds was added instead.

Another goal of the final UI was to minimize the time the user spent away from the map screen, both in time and “apparent distance.” To achieve this, the other app activities (data submission, profile management, etc.) were changed from fully separate screens to pop-up windows accessible from the map interface. The map becomes the only full screen activity in the entire app and is visible in the background during other tasks. This results in a more interactive interface, providing immediate and continual feedback. The new layout also naturally encourages users to cover a wider area. Lacking an informative layout, users might concentrate their search efforts in a single area or accidentally revisit locations. By confronting the user with a constant reminder of how their submissions are grouped relative to their current location, most users will naturally gravitate to new areas.

3.2.5 User Training

Crowdsourcing is a medium that produces reasonable reliable results when applied to tasks that do not require specialized knowledge. Burnap et al. [47] applied crowdsourcing to engineering design problems with objectively quantifiable answers to study the effectiveness of crowdsourcing for scenarios requiring technical knowledge. They observed above average results when experts within the participant base were identified and their contributions weighted more heavily. However, failing to do so negated most benefits of crowdsourcing because clusters of consistently incorrect participants cancelled out contributions from more knowledgeable persons. This suggests that raising the knowledge level of a user base should be a priority for technical crowdsourcing projects. Since net.Tagger is available to the general populace, excessively relying on a user to make technical decisions increases the probability that they will submit incorrect results. Fortunately, net.Tagger users do not need to understand most of the networking theory discussed in Chapter 2. As long as users are able to identify the infrastructure indicators discussed in 2.5 and understand the relevance of utility markings and infrastructure provider names, they will usually be able to perform accurate assessments. To train users, the app has a “Training and Examples” section (Figure 3.5) that lays out identifying information, sample images, and examples of helpful user comments for each infrastructure indicator type.

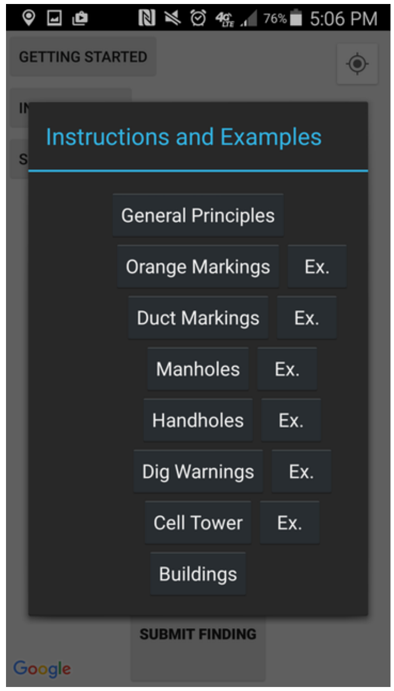

Figure 3.5: Examples Screen

Additional means of validating submissions are a priority for future net.Tagger research.

While it is inevitable that some level of user misunderstanding will lead to erroneous submissions, crowdsourcing possesses natural error correcting mechanisms. Because users can only view their own previous submissions and not those of others, multiple users investigating the same area are likely to tag the same object. The set of submissions for a single infrastructure indicator will have several that agree with each other, pointing toward the correct data. Furthermore, even if the user is wrong about their submission, the combination of an image with its GPS coordinates will be enough for researchers to extract some level of information. These redundancies reduce the level of training that most users will require for the project to collect usable research data.

3.3 Backend Services

3.3.1 Requirements

Due to the simplicity of the net.Tagger app, most web architectures and frameworks could be adapted to handle and store collected data. As for any project, the server side implementation must be reliable and secure. Finally, all components must provide appropriate GIS capabilities where needed as well as the means to maintain compatibility with other GIS projects such as OSM. Factors such as datum, map projection, coordinate systems, and time zones must be accounted for to ensure that the collected dataset can be compared to and combined with those from other sources. Currently, net.Tagger relies on technologies such as Google Maps for most of its GIS data collection and display. However, as the project eventually moves to other platforms such as IOS, net.Tagger aims to shift to open source, platform agnostic tools for tasks such as rendering. The selected architecture should be easily migrated to other tools and platforms without requiring extensive redesign.

3.3.2 Database Selection

Most GIS projects utilize an SQL-type database to store data. net.Tagger was heavily inspired by OSM and is designed to maintain compatibility with it for future research efforts, making OSM's software choices relevant to this project. While OSM does not officially endorse a specific software stack, the majority of its users, including the core OSM distribution, relies on a popular GIS add-on to PostgreSQL known as PostGIS.

PostgreSQL (abbreviated as Postgres) is a powerful Object-Relational Database Management System (ORDBMS) compliant with the SQL standards and provides many advanced features. While Postgres supports basic geometric data types, it lacks support to handle spatial data and transactions. Fortunately, Postgres is designed to be easily extensible. In 2001, the company Refractions Research released the first iteration of an add-on named PostGIS to provide basic spatial types. PostGIS has continued developing new features that not only aid data storage, but provide tools for querying and analyzing geospatial data. These capabilities extend beyond those available with more conventional GIS storage types that are limited in their ability to store accompanying metadata or large data quantities.

Most OSM users utilize PostGIS in conjunction with the OSM project's custom GIS formats, particularly the OSM XML format and its variants. The OSM XML file format is a human readable representation of OSM data. The OSM project hosts free copies of .osm files for most countries and states online, including a master planet.osm file, containing all collected data the project possesses. At the time of writing, planet.osm is approximately 50 GB of data compressed, expanding to over 500 GB uncompressed. Since plaintext XML is not an efficient storage medium, binary and compressed representations of .osm files also exist. For practical use, software packages such as the popular osm2pgsql library exist that can receive .osm files as input and insert the bulk data into a PostGIS database. The findings and metadata collected by net.Tagger are not best expressed in the table format used by packages such as osm2pgsql, as these combine most metadata into a single "tags" column that does not permit querying the individual elements. Since most of the metadata for net.Tagger such as infrastructure provider or infrastructure type must be able to be queried directly, the format is not ideal for this project. Thus, net.Tagger finds middle ground by using a PostGIS database that stores appropriate data in individual columns but keeps data such as lat/long coordinates in the same format as OSM databases. The project database is ideally suited for its specific research needs while retaining the ability to interact with other data sources through existing GIS software.

3.3.3 Scripts

Server-side processing is performed through a series of PHP scripts. PHP was chosen due its ease of deployment, preexisting code body, and user community. While PHP is considered by some to present security risks when deployed in large-scale, complex web applications, most reported PHP security flaws are not due to inherent technical flaws but poor coding practices. To rectify this, many features exist to perform sensitive processes such as password validation or database operations without requiring developers to manually implement them and risk doing so improperly. Server operations in net.Tagger are limited, primarily restricted to user credential validation, receiving GIS data and photographs, and performing database storage operations. All these operations are well-understood processes with established best practices. Because net.Tagger does not have a web presence with complicated user interaction needs, PHP is an appropriate option that fulfills the quick development time the project requires.

3.3.4 Security Considerations

net.Tagger was intentionally designed to limit the amount of sensitive data it transmits and stores. This limits the security requirements of the project to following best practices and using built-in features of its native software packages. All user submissions including profile data, tag data, and images, are sent via https POST messages utilizing Android's built-in security certificates. User sessions are recorded and authenticated via session keys in keeping with basic web application principles, and user passwords are stored in hashed and salted form. Due to a plethora of incidents where PHP developers improperly designed their own password handling procedures, PHP now automates the entire process within a single function call to store or validate a password, removing room for error. Most importantly is the decision to limit user metadata. Users are identified via a valid email address and their country of origin, limiting the cost of a potential security compromise. As a crowdsourcing operation, net.Tagger only requires the ability to track users to the extent needed for statistical metrics and the ability to recognize high contributors via leaderboard.

3.3.5 Scalability

A successful crowdsourcing operation depends by its very nature on the ability to offer its services to a variable number of users. Depending on the size of its objective, the desirable number of participants will usually be very large. OSM boasts a sizable user base, with usage statistics [27] at the end of 2015 reporting over 2.5 million registered users, with over 10,000 actively contributing data weekly and 60,000 monthly. Many of the most active users were submitting on the order of several hundred new nodes per day. Even more impressively, most reported OSM metrics showed exponential growth over a several year period. Because this thesis is intended as net.Tagger's inception, certain compromises must be made in terms of resources and scalability. Its backend services reside on a VPS that is capable of handling a reasonable number of app transactions, but would fail under the load of larger projects such as OSM. The server's resources can be scaled up to an extent, but operating at a higher scale would likely require a distributed solution. Similarly, the architecture choices described earlier place an emphasis on quick development turn-around, which does not always result in optimization for large-scale deployment. This project's choices closely mirror the archetypal Linux Apache MySQL PHP (LAMP) stack with a minor change to the database component, placing it on par with many other web-services projects. Additional improvements to net.Tagger's web services will likely accompany accompany the project's expansion. Similarly, the GoogleMaps API key that the app relies upon for generating its UI can only manage 25,000 requests per day before Google begins charging proportionately to the request rate.

net.Tagger will initially be deployed with the understanding that it will not scale in its current state. This thesis is designed to produce a proof-of-concept with limited release as part of a long-term, multiple researcher project. Aiming for a fully fleshed-out first release does not provide for feedback or course adjustments until a prohibitive amount of time and resources have been expended. Because net.Tagger is unlikely to see widespread adoption until released on several different smartphone platforms and bundled with user incentive devices, the current server backend will likely be sufficient for the near future. Any scalability issues that arise will be indicative of larger user adoption than anticipated, which would be a sign of success. They will be resolved as they present themselves through further student research projects and eventually seeking sponsorship funding after demonstrating the utility of crowdsourced network mapping.

CHAPTER 4:
Testing and Results

This chapter presents results from net.Tagger's initial release. We give overall metrics for the current dataset, analysing tagging trends by type, provider, location, and inter-event delay. Specific examples of high-quality tags are discussed, including ones that utilize net.Tagger's unique capacity to capture low-permanence infrastructure indicators. We demonstrate tag validation through Google products and manual image inspection, categorizing submissions by accuracy for future research. Finally, we discuss examples of erroneous net.Tagger user submissions, including methods for identifying errors and extracting useful information from incorrect tags.

Since the proposal stage of this thesis, its primary focus has been providing a working proof-of-concept app/server framework. Because crowdsourced network mapping is a largely untested concept in the larger research community, much of the net.Tagger project thus far has been aimed at identifying target data and refining the collection process. Section 2.5 discusses the results of the former, and Chapter 3 describes the latter. However, even though this project's primary goal is not data collection, a discussion of its preliminary results is still relevant to demonstrate the utility of the net.Tagger implementation and show what analysis will be possible after its future widespread release. Another valuable set of results comes from our initial user community's experience. Feedback on the user's experiences provides metrics about net.Tagger's usability and whether portions of its design enhance or detract from gathering useful data.

4.1 Initial Release

While net.Tagger's eventual goal is to infer physical network topology, this requires a fairly complete tag set of a given geographical area. Time and resources did not permit an app release on a large enough scale to accomplish actual topology mapping. Without complete coverage of an area, it is difficult to state whether a series of tags demonstrates a unique underlying network feature or if further mapping of the surroundings would show a uniform distribution of more tags without useful trends. At the time of this writing, net.Tagger is still in its beta testing phase, and the main intent of this limited release is identifying and correcting performance and stability issues that did not present during development. Releasing only to family members, Naval Postgraduate School (NPS) students, faculty, professional colleagues, and friends with a clear description of the project's current status increases the likelihood of helpful user feedback. Skipping this step and pushing the app to as large an audience as possible without a smaller initial release would likely end in many of the target users discovering net.Tagger, experimenting briefly, and then uninstalling the app out of frustration over its unpolished appearance and function.

Overall statistics for the project at this time are as follows:

Table 4.1: High-Level net.Tagger Statistics

Copies Distributed	25
Profiles Created	12
Contributing Users	9
Total Tags	166
Tags w/ Image	101
Total Providers	18
US States Represented	5
Countries Represented	2

The two most common reasons we received from 13 users who declined to participate were "insufficient personal time to participate" and "no IOS version of app." The following figures display trends of the 9 contributing users. Figure 4.1 parallels similar projects analyzed in Section 2.4.

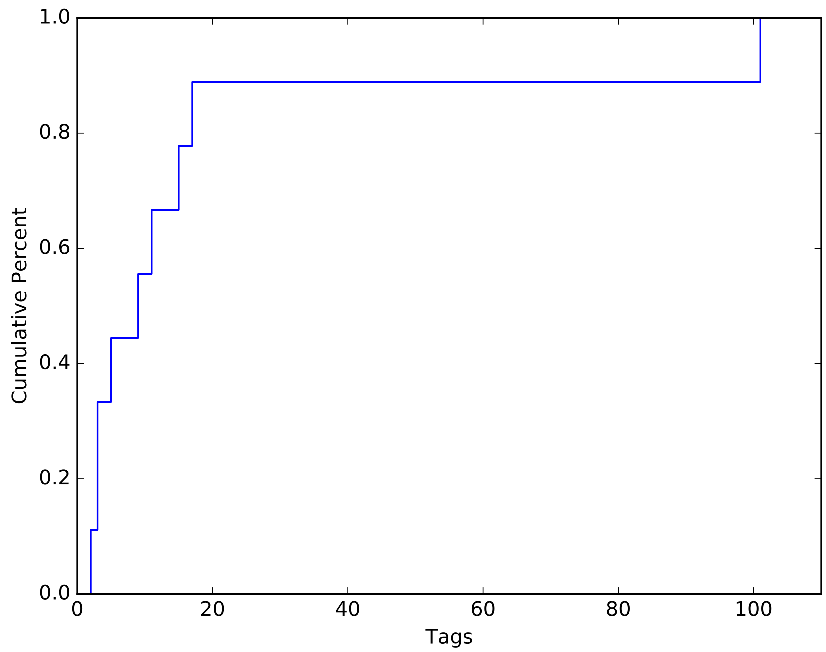

Figure 4.1: CDF of Tags by User

Even with a small sample size, a trend is clearly visible where a large number of users accounted for a small portion of the total tags. Conversely, a small number of users contributed the majority of the tags. Out of 166 tags, the top three users submitted 133 tags, with 101 the highest number. Presumably, when net.Tagger scales up in size, this trend will continue. Assuming rough equivalence with OSM use rates, we can anticipate most tags coming from a core 5–10% section of users, with the rest of the user base submitting at lower rates.

In Figure 4.2, we examine the number of distinct types (manhole, duct, etc) of infrastructure tagged per user.

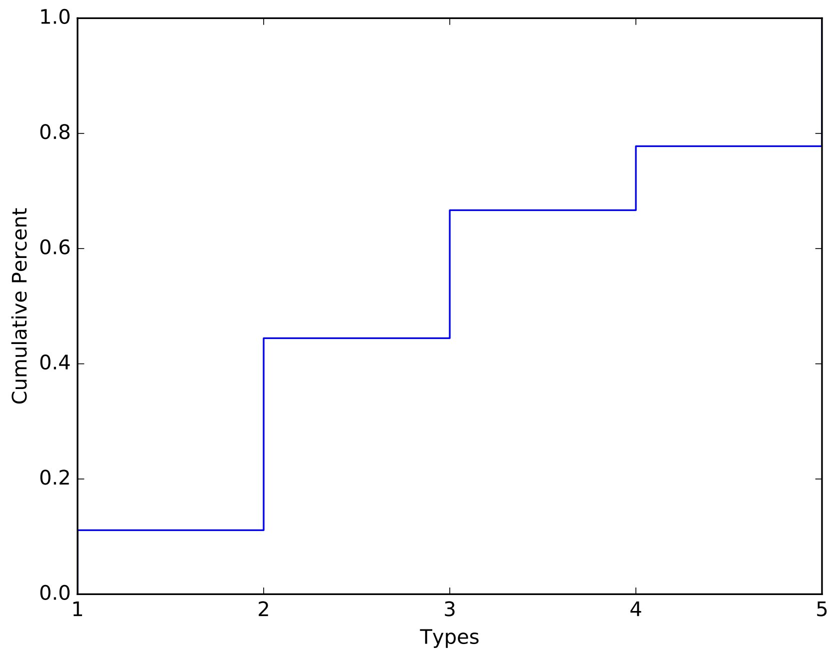

Figure 4.2: CDF of Infrastructure Types by User

The maximum number of infrastructure types was 5, which 11.1% of our users reached. In examining this metric, we seek to determine whether some users tag only one type of infrastructure (perhaps because of where they live, or what they commonly notice), or are adept at tagging many or all of the types of infrastructure in which we are interested. We observe a generally uniform distribution of infrastructure types, suggesting that our user base does not exhibit any particular bias in the tag types. In Figure 4.3, we examine the number of different infrastructure providers in each user's set of tags.

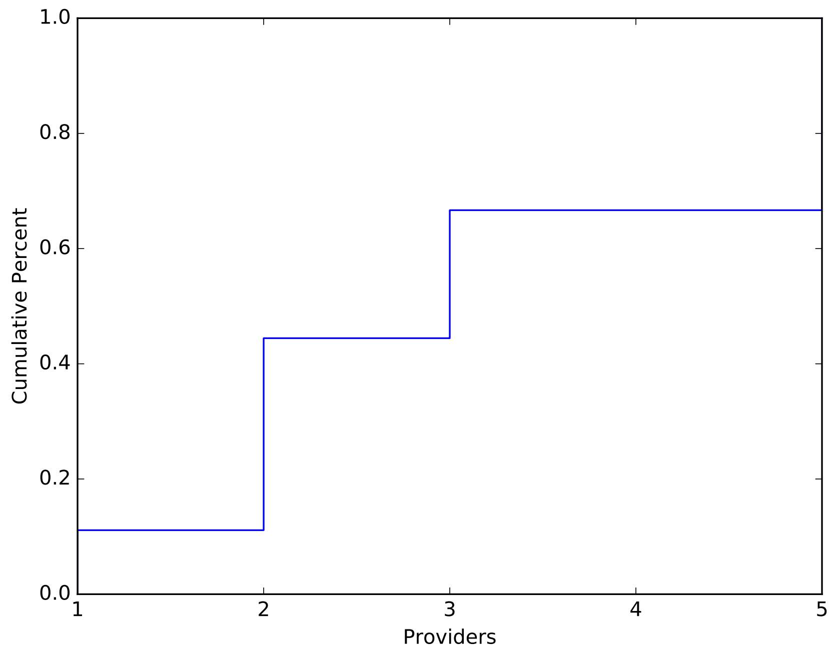

Figure 4.3: CDF of Infrastructure Providers by User

Users were able to choose from six major providers, "unknown," or an "other" option where the user notes the name of the provider in their comments. The six specific providers were selected based on informal analysis of the most common providers encountered during initial fact finding research, with the intent of expanding and tailoring the app's options in future releases. Of the eight available options, the maximum number of providers was five, achieved by 33.3% of users. Every user who submitted more than 10 tags fell into this category. This result implies that users who contribute beyond a certain minimum threshold will encounter a diverse set of providers, even if they limit themselves to one geographic location. Fully 88.9% of users submitted at least one "other" tag, specifying an additional provider. A further 66.6% of users submitted at least one "unknown" provider tag. In Figure 4.4, we examine each user in terms of how many zip codes they submitted tags from.

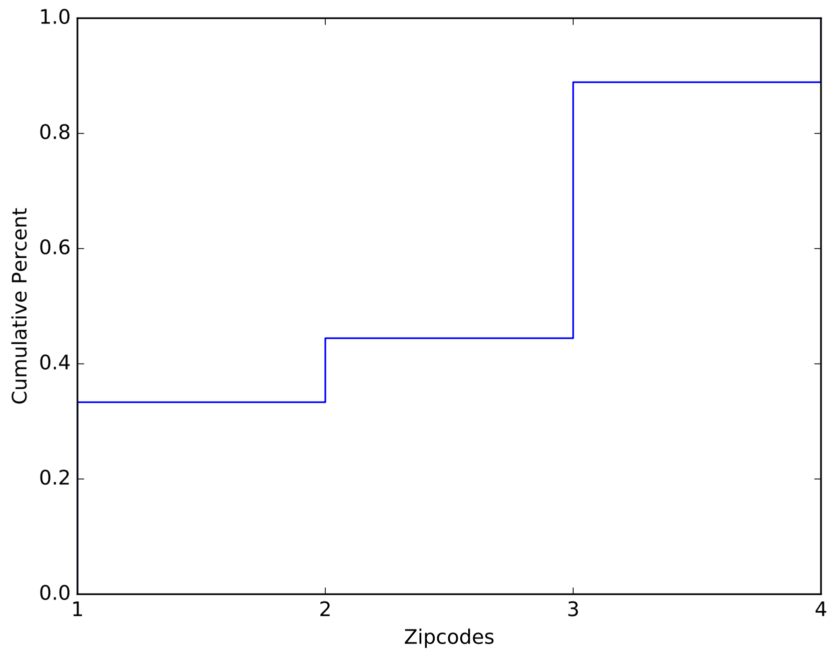

Figure 4.4: CDF of Zipcodes by User

Although zip codes are defined and modified due to multiple metrics in addition to geographical zoning [48], they correspond to location and population distribution, providing a useful approximation of a user's tagging locations. Google's Geocoding API [49] provides a reverse geocoding lookup feature that we utilized for this analysis. The service requires crafting of simple HTTP requests with tag Lat/Longs as URL parameters to return Javascript Object Notation (JSON) data including a zip code with suffix, which we automated to simplify analysis. The maximum number of zip codes for an individual user was four, which 11.1% of users achieved. The same number of users only submitted from one zip code, with all others visiting two or three. This indicates that even users with a small number of tags will still exhibit some level of geographical diversity, while still remaining relatively local.

Overall, infrastructure providers, types, and zipcodes all showed fairly uniform distributions. This might suggest that the variety of providers and tag types scales up as users expand their geographical area of coverage. However, the sample size is too small to be conclusive at this point.

Figure 4.5 shows per-user delays between sequential tagging events.

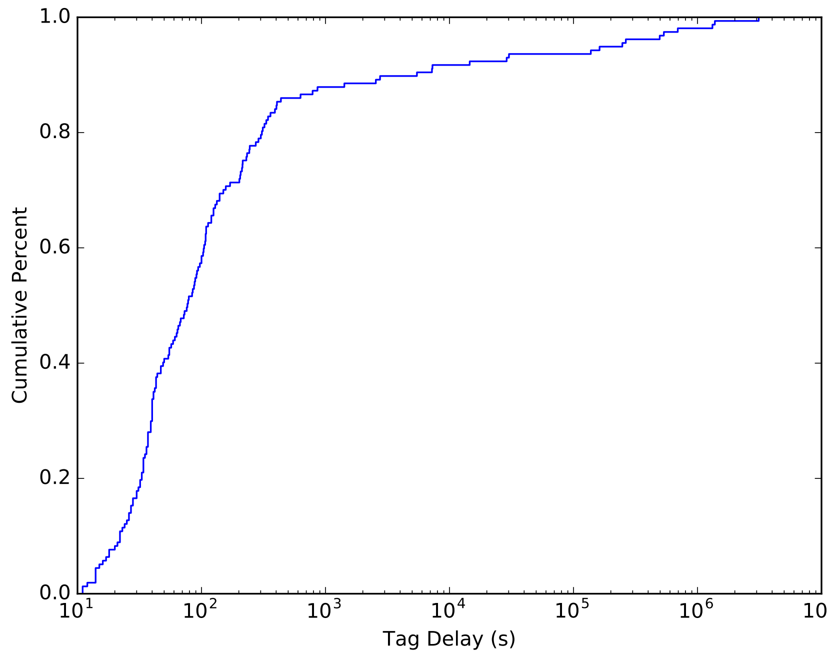

Figure 4.5: CDF of Tagging Delay

It suggests that most users submit tags in relatively rapid succession of several minutes between tags and then are inactive for several hours or days. This demonstrates one method of use envisioned in Section 3.1 of users allotting dedicated periods of time to tagging instead of making periodic submissions over a larger period of time. Most users at this time are gathering evidence by direct request of the net.Tagger team, which likely takes the form of dedicated tagging trips. Another possible explanation for this trend is that upon seeing a possible submission, users become aware of other possible tags in the area, temporarily increasing their vigilance. If future research confirms this hypothesis, some type of user notification when entering high-density areas might provide a similar effect. This idea is more thoroughly explored in Chapter 5. User submission periods might become more regular as the net.Tagger community grows and community incentives are introduced. Further research is necessary to determine if these inferences of behaviour are correct or if current conditions of data collection are artificially introducing them.

4.2 Quality Examples

Some of our 166 tags serve as examples of ideal net.Tagger submissions by combining multiple indicators. They provide extra context of their surrounding areas even if the location has not been exhaustively covered by net.Tagger users, permitting preliminary inferences about underlying network topology.

The following submission images are presented with their verbatim database extract, representing the sum total of information available to us about a specific tag. Fields containing Personally Identifying Information (PII) are censored in this section for privacy reasons. Entries observe the following format:

Table 4.2: Database Entry Format

Tag ID	TX ID	User ID	Lat	Long	Timestamp	Provider	Type	Comments

Figure 4.6 combines three features in one: a duct marking, a "telephone" manhole cover, and an orange "COMM VAULT" marking.

(XX,A3746D62E7381E3D4141B903CEBFC5C0FB39DC20,XXX@XXX,XX5892028,-XXX5903990,"2016-02-08 16:54:36-05",Unknown,Manhole,"Possibly AT&T")

Figure 4.6: Communications Vault with Duct

Even though networking equipment is not specifically referenced, FOCs carrying network traffic are frequently co-located with phone lines due to the high expense of laying new ducts. The markings and manhole access indicate some sort of central node, and the duct marking gives context about how it connects to other nodes.

Figure 4.7 demonstrates a desirable net.Tagger datapoint by combining FOC ducts with a building of some sort.

(91,C3ADC0DA3F8E36E09E67BC636AED99DD5F654505,XXX@XXXX,XX5893242,-XXX5906732,"2016-02-08 16:56:20-05",Unknown,"Orange Marking (misc.)","Possibly AT&T")

Figure 4.7: Duct with Building

The user did not tag the building separately and likely did not identify the potential utility of doing so, however their tag image shows the connection. It is possible that the duct simply passes under the building and the two have no relation, but their association increases the likelihood that the building houses some type of networking equipment. net.Tagger researchers would flag this as a location of interest and monitor the area for other tags indicating additional FOCs or access points, looking for clues that the structure is a local nexus of networking infrastructure.

4.3 Low-Permanence Indicators

A unique capability of net.Tagger is its ability to capture infrastructure indicators with low persistence. While other mapping projects described in Chapter 2 target large, static networking features such as railroad ROWs, net.Tagger can capture infrastructure with relatively short-lived indicators when users are in the area tagging. Such "low-permanence" indicators primarily concern FOC cables and ducts, which are valuable mapping data for connecting network nodes. Because they are marked with chalk or street paint, FOC markings exist for short amounts of time, but are much more likely to indicate current information than other indicators. Figure 4.8 and Figure 4.9 show examples of this phenomena.

Figure 4.8: Orange Marking and TV Pedestal, Bark and Grass

Figure 4.9: Duct Marking, Grass

Because these examples are placed over grass and bark dust, they possess a low persistence and will soon disappear from sight. As net.Tagger's community increases in size, its ability to capture temporary indicators will correspondingly grow as well.

4.4 Tag Verification

Because the initial net.Tagger release only featured 12 users (9 actually contributing) spread across 5 states, there were no cases of two users tagging the same finding. However, a number of submissions were at least partially verifiable by searching the tag Lat/Long on Google Earth and trying to match results against the user submitted tag image. This approach is potentially time consuming, as it requires manual human validation for each tag and is not always successful if the target is out of sight from the Google Earth/StreetView reference point. Because of these complications, manual verification would only be employed on a case-by-case basis by net.Tagger researchers who identified certain tags as highly relevant for area-specific inferences. Despite its shortcomings, we successfully employed manual verification for both urban and rural locations to prove its utility. An urban example of this process involves a tag in downtown Cambridge, MA. Figure 4.10 shows the image submitted by the user, which, as a manhole stamped with "Communication," appears to meet all criteria of a good tag.

Figure 4.10: User-submitted Image

If net.Tagger researchers believed verification of this tag was necessary before relying on it for further inferences, it can be investigated via Google Earth's StreetView feature. Figure 4.11 shows the overhead view of the tag's coordinates on the left (Marker #113) and the StreetView on the right.

Figure 4.11: Google Earth at Image Coordinates

Even at a lower resolution, several manholes are clearly visible that appear to match the user tag image in 4.10. While not as conclusive as a matching tag from another user, at least partial confirmation of the tag has been made.

This approach can even work in rural areas. One user submitted two tags within minutes of each other in the middle of a forest on the Monterey Peninsula. The user indicated a cell tower (Figure 4.12) and Pacific Bell handhole near each other in an area away from all other structures except a construction site.

Figure 4.12: Cell Tower, User Submitted

Because cell towers often connect to adjacent FOC lines, the combination of a tower and handhole in a more remote area is an important finding. When interviewed, the user confirmed this finding, stating that he discovered the tags while trail running. Even if the user had not been available for comment, Google Earth can still provide initial confirmation.

Figure 4.13 shows the Google Earth coordinates of the cell tower tag (Marker #103) and Pacific Bell handhole (Marker #104).

Figure 4.13: Cell Tower, Google Earth

Although low resolution, Google Earth clearly shows the cell tower's profile rising out of the forest in the exact location that the user's image and tag places it. It is not possible to make out the handhole, but verifying one submission increases the chance that another tag from the same user several minutes later is valid as well.

An additional verification process that focuses on the tag's specific traits instead of its location involves checking the user's description of the item against the user-submitted image. This is only possible if the user chooses to submit an image with their tag, which will eventually be incentivized as discussed in Chapter 5. Infrastructure provider, type, and comments can all be vetted against a submission image by a net.Tagger researcher in a brief amount of time and the tag reliability ranked accordingly. For this thesis, we ranked tags against their images according to the following categories:

- All data fields concurred with image. In Figure 4.14, the infrastructure type and provider are clearly visible and concur with the user's form submission.

('XX', '5EDDE570778C03D96FD378CBF012853BDAEA3309', 'XX@XXXX', 'XX5545949', '-XXX6765036', '2016-02-14 10:47:03-05', 'Bell', 'Manhole', 'null')

Figure 4.14: Bell Manhole

Although the user could have clarified “Bell System” in his comments, the tag entry is still complete and contains no misleading or incorrect information. Submissions in this category are confirmed by their images. In our initial dataset, 77 of 101 image submissions fell into this category.

- Some data fields are incorrect, however the image contains enough information that any errors are immediately apparent. Figure 4.15 shows a submission described by the user as a handhole operated by an unknown provider.

('XXX', 'EAF724412CD9EC5D3456D5924CF42AB5366D32E7', 'XXX@XXX', 'XX5757070', '-XXX9336365', '2016-03-02 16:53:21-05', 'Unknown', 'Handhole', 'null')

Figure 4.15: Mislabeled Manhole

A cursory inspection of the image shows a manhole instead of a handhole, which the user has misidentified. However, the discrepancy is immediately apparent, and the tag can be quickly updated by net.Tagger researchers with no loss of information due to the error. The image even contains enough resolution to zoom in and read the inscription “Bell System,” meaning that researchers can even fill in the user’s blank provider field. Submissions in this category are confirmed, corrected, and potentially improved by their images. In our initial dataset, 11 of 101 image submissions fell into this category.

- No discrepancies between data fields and the image are visible, however the submission form data contains information not verifiable by the image. Tags in this category are more complicated to categorize. The difficulty arises from the fact that net.Tagger researchers do not know whether the extra information in the form is due to factors not visible in the image, or represents a user error. Figure 4.16 shows a submission where the user identified an orange marking and specified “Comcast” as the provider in the tag comments section.

(XX, '75295219C09E3A5884AB85F5A3121E11D86A9607', 'XXX@XXXX', 'XX7180402', '-XXX6330881', '2016-02-29 17:24:18-05', 'Other (note in comments)', 'Orange Marking (misc.)', 'Comcast cable')

Figure 4.16: Indeterminate Orange Marking

The image clearly shows a duct marking, indicating that the user partially identified the correct infrastructure type. However, the user's rationale for submitting Comcast is not readily apparent. Normally, provider information for an orange marking would be painted on the ground or not marked at all. Because the image does not include the provider name in the marking, the submission raises the question of whether the user knows something not included in the image, or if the user is mistaken. net.Tagger researchers possess a large enough sample set to determine fairly accurately from a well-taken image what information is or isn't available, and this image seems to lack the information a user would need to accurately specify a provider. After reaching out to the user, we determined that the marking led to a residence serviced by Comcast, thus the submission was accurate. If this additional validation step was not available, the apparent discrepancy between form data and image would have forced net.Tagger researchers to partially downgrade the submission, keeping the infrastructure type but classifying the provider as "unknown." Submissions in this category might be partially invalidated by their images, but still contain some useful information on a case-by-case basis. In our initial dataset, 6 of 101 image submissions fell into this category.

- The image contains enough information to determine that the submission does not represent a valid net.Tagger data point. A detailed treatment of this category is given in Section 4.6. User submitted images provide the most reliable means to vet net.Tagger data through this process. In our initial dataset, 7 of 101 image submissions fell into this category. It is important to note that these erroneous submissions are not necessarily due to user incompetence or a misunderstanding of net.Tagger principles. Users are subject to their own time constraints while participating and are not expected to be subject matter experts. Many of our test users expressed concern about potentially submitting erroneous data, and we assured them that providing images along with their tag data would give the net.Tagger team the means to vet their finds. The limited scope of this project allows us to tightly control more variables than a full-scale release; a feature we took advantage of by instructing our users that when in doubt about a finding, they should submit anyway. This helps fulfil one of this project's objectives by revealing the ability of an average user to correctly identify Internet infrastructure.

4.5 Tag Comments

Much of the need for projects such as net.Tagger comes from the large variety of competing and overlapping telecommunications providers who communally own and operate the Internet backbone's infrastructure. As different corporations change ownership, merge, acquire new assets, and lease infrastructure to others, the infrastructure indicators targeted by net.Tagger have the potential to become increasingly obfuscated. The "comments" section of a net.Tagger app submission is of critical importance to augmenting a tag and mitigating data gathering challenges. Even minor or incomplete tag comments can give net.Tagger researchers insights into the validity and relevance of a given tag for making further inferences. The more tags a user submits, the more likely he or she will begin to build a picture of what infrastructure indicator trends exist in their local area, and which of their findings are unique or relevant in a broader context. Ideally, as the net.Tagger user base grows and matures, tag comments will grow in importance and usefulness. Even in net.Tagger's current phase, tag comments are an important tool to fill in information gaps not covered by the app's dropdown options in the data submission screen. Putting too many options in a menu clutters the UI, removing a user from the submission cycle. Once the net.Tagger dataset is large enough, the app can be modified to offer a location aware selection that offers a user the most prevalent providers in the area to choose from. This can be further combined with on-device caching of the users' past submissions to simplify the submission process for users on an individual basis. Tag comments can not only clarify submissions, but provide additional data sources for net.Tagger researchers to mine for possible app improvements.

As an example of tag comment utility, different telecommunications providers such as AT&T and CenturyLink own or operate part of the historic Bell System, often as independent entities. Listing all these possibilities in the app's data submission screen would likely lead to user frustration. However, our initial findings showed that most users will still clarify which specific Bell iteration they have discovered in their comments. Out of the 35 tags users labelled as "Bell," we received comments clarifying "Pacific Bell," "Bell Telephone," "Pacific Telephone," and "Bell System." This amplifying information is useful for determining local provider trends and isolating specific infrastructure features.

Unfortunately, many tags in the initial net.Tagger dataset either lack comments or do not contain enough information to be useful. Of 166 tags, 51 did not include any, representing approximately 20% of all submissions.

4.6 Errors and Noise

Preliminary interpretation of the 166 tags at the time of this thesis shows a number of complications. Because of the close ties of active users to the net.Tagger team, reaching out for clarification was much more straightforward than with a general public release. This offered a temporary advantage in determining if a submission was truly erroneous or only appeared so based on the data available. For example, the following tag (Figure 4.17) was submitted from downtown Monterey:

(136,AEEA6C4CA37F9BB9CC9CF78901C39EE37AF80D04,XXXX@XXXX,XX5984008,-XXX8957686,"2016-03-06 16:02:15-05",Unknown,"Duct Marking","2-4""ducts")

Figure 4.17: Duct Marking Tag

The user's data entry indicates a sidewalk duct marking, annotating the marking's text in the comments section. However, viewing the image submitted with the tag shows a duct marking that appears to be drawn in red paint, which by APWA standards would indicate electrical power instead of telecommunications equipment. Under the information available between the tag entry and attached image, net.Tagger researchers would likely conclude that the user mistakenly submitted a power cable duct as as telecommunications asset, requiring reclassification of the tag as inaccurate. However, after discussing the tag with its responsible user, we concluded that he could properly identify PMS 144 Orange, and local lighting conditions caused his smartphone camera to misrepresent the color of the markings.

Other submissions (Figure 4.18) were clearly erroneous, however verification was straightforward because the users were careful to provide details in their comments.

(86,524B8FABBE717B5AACCEFC4383BE6B82176B8865,XXX@XXXX,XX5796949,-XXX6177637,"2016-02-05 14:40:39-05","Other (note in comments)",Manhole,"PacfiCorps electrical vault")

Figure 4.18: Electrical Vault Tag

This submission came from a user lacking a networking background. Upon analysis, the image lacks positive indicators of networking equipment, and PacifiCorps is a utility company that does not provide telecommunications services. When interviewed, the user stated that he was unsure about the find, but chose to submit with as many details as possible to facilitate eventual verification. Vetting the tag was simple for the net.Tagger team, and the same user submitted a number of high quality tags in the adjacent area.

Other erroneous tags (Figure 4.19) did not have additional comments, but could still be downgraded in reliability due to the image.

(112,855D69370A5AE4B5E8375091D85A98781B3C43C4,XXX@XXXX,
XX3627900,-XX0911454,"2016-03-03 16:21:44-05",Qwest,Manhole, null)

Figure 4.19: Qwest Manhole Tag

This submission was marked as a Qwest manhole with no amplifying comments. The manhole bears the engraving "BECo," which according to low validity sources [50] is the marking for Brooklyn Edison Company, a power utility company based in New York City. Based on the conflicting tag data/image information, this data point does not possess enough reliability to be used for future inferences without more information.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Future Work

Some research projects complete their investigations and list "Future Work" ideas as an afterthought with minimal content. Because this thesis represents the first effort in creating the larger net.Tagger initiative, this chapter takes on significant importance. While the net.Tagger project has a clearly defined goal – broad mapping of physical network infrastructure through crowdsourcing – the specific implementation and requirements continue to be refined. Implementing an initial mobile app and server framework, performing data collection, and gathering user feedback allowed us to identify additional features and project enhancements that will greatly increase the quality and utility of research findings going forward.

This chapter addresses four categories of future work for net.Tagger. A primary area of work will involve additions and enhancements to the smartphone app, including porting to other platforms, enhancing the UI, and increasing the map overlay to include the entire project dataset. Second to be upgraded is the backend server infrastructure. This includes a full security audit, better web services handling, and integration with the OSM stack and dataset to perform native map renders. Third, data analysis and data fusion will greatly enhance the research value of the project dataset. Finally, and most importantly for net.Tagger's expansion and future, is development of features and incentives to increase adoption and use.

5.1 App

5.1.1 User Interface

While the UI has undergone considerable evolution over the course of this project, it is still a product of the short development timeframe. Due to the increasing quality of most smartphone apps, potential users are likely to view the quality of new apps as a function of visual presentation, workflow intuitiveness, and overall ease of use. Even if UI features do not directly increase the quality of collected data, they are still important to net.Tagger's success as a crowdsourcing project. An intuitive user experience will provide fewer entry barriers to users, particularly those lacking a technical background who might be intimidated by a less usable set-up. The main display can be improved through implementation of minor features such as using slide-out menus instead of static buttons, which crowd the display when not being used. Multiple beta testers were concerned about their ability to submit accurate data. Because these users were people already possessing technical backgrounds, their concerns indicate that the entire spectrum of potential users would benefit from additional in-app resources guiding the submission process. One feature accomplishing this is a tutorial style walkthrough offered to users upon a fresh install. Many apps provide a demonstration of this sort, since static "Help" documentation does not always translate into practical understanding for all users. Some testers reported an initial hesitancy to begin tagging for fear that they would make a mistake and pollute the project database with false information. In addition to a walkthrough, another feature that would ease their concerns is the ability for users to delete their own tags if they feel the submission was errant. Currently, there is no delete mechanism in the app. However, multiple testers inadvertantly submitted tags when first exploring the app and voiced concern that they could not clean up their mistakes. Giving users the power to delete tags allows them to experiment without fear of messing up, not only reducing research errors but also shortening the time between installation and feeling comfortable about participating. For research purposes, deleting a tag in the app should not actually delete the information from the net.Tagger database. Knowing how frequently users delete data relative to time spent using the app is a useful metric for researchers. If multiple users submit and delete tags in a specific location, this could indicate that an infrastructure indicator exists but is ambiguous and needs further validation before using it for network inferences. Instead of actually deleting the submission from the net.Tagger database, the in-app delete option should flag the appropriate database entry, remove the marker from the user's map, and display a user message that the tag is deleted. This provides the user with assurance that the net.Tagger team knows of the error while still preserving the data for other purposes. Combining a tutorial walkthrough with tag deletion capability will ensure that users feel more confident about participating while increasing the likelihood of correct submissions.

In addition to app UI improvements that will help users get started with net.Tagger, other planned features will assist users while gathering data. One feature suggested by test users is automated notifications once the user enters a new area with few or no tags. Users will be able to enable this feature from a settings dialog and configure it to define what a “new area” consists of. For example, a user could set their net.Tagger instance to alert them if they are more than a certain distance from any of their past submissions. Another alert might trigger if the user is near an unverified tag from another user, indicating nearby targets of opportunity. Not all users will desire a notification feature, and it is of little utility for users during dedicated tagging sessions who have the app open where they can actively see the map. However, other users might be interested in submitting intermittent tags while they are performing other tasks, and would appreciate notifications informing them that they are in a potential tagging location. The notification feature can be further integrated with the upgraded map display to display helpful messages to users when it triggers.

5.1.2 App Backend

In addition to the UI improvements of Section 5.1.1, some improvements to the app’s backend are necessary before undertaking broader distribution efforts. Of primary importance is improving the app’s location sensor routines, which define the precision and regularity with which the app samples the user’s GPS coordinates. Currently, the app uses manually coded location routines that use fine-grained Android functions instead of more granular API methods. These provide the high accuracy necessary for accurate tag measurements, but place an unreasonably high load on the smartphone’s battery life. Android developer guidance recommends using native location tools available as part of Google Play service APIs, as they automate these processes to optimize battery life without compromising location accuracy. Unfortunately, net.Tagger cannot make use of them until the app is registered with the Google Play Store. Once they become available to the project, refactoring the app’s code to use them will provide better battery usage, reducing the potential for users to become frustrated with the app. Market research surveys of app users [51] identifies battery issues as a motivating factor in users giving negative reviews or uninstalling apps, particularly with mapping applications. This gives net.Tagger incentive to use all available resources to manage app resources well. Other smartphone sensors discussed in Section 2.6.2 can be leveraged to improve research data without requiring active user action. The Android orientation sensor can be used to directly calculate the orientation of a device relative to magnetic north, however it requires substantial processing power and has been deprecated since Android 2.2 [52]. Android provides methods that calculate equivalent results without utilizing raw orientation sensor data. Another capability that can be leveraged futher is the GPS sensor. Currently, the app only transmits a lat/long and blocks users from submitting if the GPS sensor's calculated accuracy is less than 30.0 meters. Instead of setting an accuracy limit, the app will transmit the sensor's accuracy at time of submission. The combination of lat/long, position sensor accuracy, and device orientation for each tag will provide a much more accurate tag than lat/long alone.

Another useful capability to implement would be the ability to store user submissions on the smartphone if network services are not available, permitting users in remote locations to tag findings for upload once service is restored. This feature would require careful implementation and configurability from the user's settings menu. Mismanagement could place a burden on device storage and mobile data, particularly if the user accumulates a large number of findings before reentering a service area. These issues could be addressed by allowing users to place storage limits on the app and limit burst transmissions to when the phone is connected to wifi networks. This would function similarly to smartphones that avoid downloading app updates until connected to wifi, preventing excessive mobile data consumption.

To facilitate future software development, the net.Tagger app should continually improve its error handling and crash reporting. Currently, the app utilizes the Application Crash Reports for Android (ACRA) library, which automatically sends stack traces and phone version information to a net.Tagger server upon full crashes. This proved very useful during the initial app release, when almost half of net.Tagger users experienced unrecoverable crashes during installation. ACRA crash reports quickly narrowed the problem to Android Version 6 smartphones, which utilize a radically different permissions model than versions used during development testing. Once identified, the issue was quickly patched and a new version pushed out. While these reports are invaluable, they only provide information when the app experiences a complete crash, which should occur less frequently as the production code evolves to better anticipate error conditions. These improvements come at the cost of less information to troubleshoot issues. Even if the app handles errors without full-on crashing, different features may still not be functioning as intended. While coding and testing, net.Tagger developers can make use of debugging features such as Android Studio's

LogCat to view helpful messages about the app's state. Before large-scale distribution, net.Tagger should implement improved logging systems to send relevant information about experimental or high probability of failure processes to net.Tagger servers. Unlike now, a full-scale release does not offer the ability to reach out and contact users about their issues as readily, and automated processes must be put in place to collect relevant information.

5.1.3 Distribution

A successful crowdsourcing project relies on effective advertising and providing a simple way for potential users to obtain and install the app. Currently, the net.Tagger app exists as an .apk file download on a Center for Measurement and Analysis of Network Data (CMAND) website. This approach requires users to visit the website, manually download the .apk file, disable their smartphone's security protections against third party unverified apps, and finally install the app. While sufficient for initial beta testers already associated with CMAND, this implementation is not suitable for wider distribution. The next logical step is signing, registering, and importing the app into the Google Play Store. In addition to increasing net.Tagger's profile to its potential user community, most smartphone owners will not trust anything outside of official distribution channels, and release through the Play Store removes many security concerns users might have with a third party app. Also important to the project's success is the ability to push out updated versions of the app to users as the improvements described in this chapter are implemented. Hosting the app as a file download on the net.Tagger website requires users to download fresh copies every time a release is made. The effort this entails reduces the likelihood users will perform the extra step, hindering the project's ability to grow and expand. Integration with the Play Store gives project developers the means to release updates with a far greater certainty that users will receive and automatically install them. The Play Store also provides users with the means to assign numerical ratings and reviews of apps, which gives net.Tagger another source of feedback. While a useful asset, Play Store feedback also increases the importance of identifying and removing as many bugs as possible before release, as bad initial reviews could discourage potential users from installing. To this end, net.Tagger should ensure compliance with Android's published series of quality control guidelines [53] before app release. Once better mechanisms of distribution are in place, net.Tagger can take advantage of additional resources to more broadly advertise the project. Resources such as the North

American Network Operators Group (NANOG) Mailing List or OSM forums can be used to both increase project visibility and solicit feedback.

5.1.4 Platform Porting

Currently, net.Tagger only exists for Android devices. The Android development community provided many useful features and resources that were a key factor in producing a usable prototype within the time constraints of this project. However, limiting net.Tagger to Android would neglect the sizable market share of potential users who use other smartphone platforms such as Apple's IOS. In late 2015, IOS represented approximately 28% of the US market share, second to Android's 67% but well ahead of Windows' third place 3.5% [54]. Technologically, it is not possible to port or cross-compile net.Tagger's java-based Android code directly to IOS's Objective-C. However, the UI design, workflow, and server infrastructure can be reused, amortizing the cost of design and testing of these components. Instead of writing the IOS app from scratch, it can be built to an existing specification and template, thereby presenting fewer challenges to an experienced programmer.

5.1.5 Map Display

Currently, the net.Tagger app main screen displays the individual user's submission history in the form of markers placed on a Google Map overlay. The app accomplishes this by keeping a local data file holding their past tags in the app's private directory. Every time the user submits a tag, the file is updated and the map reloaded to enter the marker. Although the data file can store many different types of data, the only information currently stored is a tag id and lat/long for each submission. The main advantage of this approach is that it requires no management of a distributed dataset. Each user's smartphone maintains a local copy of its history while sending more detailed submission reports to the central server. A more ideal app configuration would display markers representing the majority or all of the net.Tagger dataset to indicate areas that have already been searched. Users should be able to set a variety of display filters on their map, including displaying all tags by all users, all tags by the smartphone's owner, all unverified tags, and tags by indicator type. This will allow users to scale back their display if app performance and mobile data are an issue, as well as assisting users conducting searches to target specific leaderboard categories. This would permit users to investigate existing findings to perform verification tags or avoid them in order to search for original findings. Including this feature will require additional network functionality and careful consideration to avoid burdening users' smartphones. Other applications such as Google Maps also allow users to tag features such as gas stations and restaurants. However, implementing this in net.Tagger will require extra caution due to substantial amount of data that must be pushed to users in areas with a high infrastructure indicator density. With careful planning and scheduling of data pushes to users, net.Tagger will be able to provide a dynamic, informative display to its users without incurring performance or data consumption issues.

5.2 Server

5.2.1 Security Considerations

net.Tagger was intentionally designed to limit the amount of sensitive data it transmits and stores. User submissions including profile data, tag data, and images, are sent via https POST messages utilizing Android's built-in security certificates. This delegates the security of sensitive data in transit to existing security implementations, providing a higher level of security than creating custom net.Tagger transmission protocols. A more likely risk comes from a breach of data residing on the net.Tagger server. Instead of the convenience of built-in methods for the app, the net.Tagger server must host and secure multiple web and database services while ensuring their availability for all required processes. The simplest means of securing data at rest on the net.Tagger server is to refrain from storing data that requires securing. A user profile only contains a nickname, email address, country, and password. The only information intended to be uniquely identifying is the email address, which is used to distinguish users for research purposes, and the nickname, which will be publicly available on the leaderboard once implemented. This reduces both the potential consequences of a data breach as well as the likelihood of attackers viewing net.Tagger as a worthwhile target. However, this does not eliminate the need for the net.Tagger research team to protect PII entrusted to them by the user community. Because of the tendency for people to reuse passwords and email addresses when registering for web services, access to the four components of a net.Tagger profile could give attackers information useful for targeting users on websites unrelated to net.Tagger.

Limiting user data reduces security requirements of the project to following best practices and using built-in features of its native software packages. net.Tagger backend components such as Apache and PHP have established security practices dictated by their own [55] or third party foundations [56] providing guidance that is sufficient to secure most simple web applications using their products. Basic security precautions for net.Tagger are in place, such as storing user passwords in the profile database after hashing and salting with PHP's native password handling features. However, because of this project's short development time, a full security audit of the app and backend server is still pending.

Any audit will have to take into consideration three possible attacker objectives: data theft, data corruption, and service interruption. Data thieves would target user profile or tag data. Both types of data include database entries, with tag data also including seperately stored image files. Tag images would be of little utility without the accompanying database entries to correlate them to users and locations, so any data theft attacks would involve some form of database attack.

Data corruption attacks would attempt to either delete and corrupt data stored on the server or insert false data points. Instead of exfiltrating data, these adversaries actively seek to modify data on the server. While more disruptive, modification attacks are harder to execute against the net.Tagger server because most of them would require some form of superuser permission. The PHP scripts that interface between received tag data and the databases do not have modification or delete database privileges, which exist only for the postgres superuser.

An attacker could attempt to craft fake tag submissions, which are simple HTTP POST messages carrying JSON data and could be easily replicated. However, the server scripts will not accept submissions without a valid session ID from an app instance, which can only be generated by submitting credentials that match profile entries on record in the database. Even though corruption attacks may be more difficult to launch, the security audit should still ensure that all Apache, PHP, and database instances are locked down to reduce their likelihood of occurring.

Finally, service interruption attacks would attempt to deny net.Tagger server availablility through some form of Denial of Service (DoS) attack. These adversaries could perform large numbers of web requests or make net.Tagger submissions that do not require sessions credentials, such as submitting profile data to fill up the database. Although there are limited remediations against these attacks, an audit could ensure that the net.Tagger server has enough scalable resources available to adapt to any DoS attempts.

5.2.2 OSM Integration

net.Tagger is heavily inspired by the OSM project and will likely draw upon the OSM software stack and dataset for future work. Because of OSM's open source philosophy and licensing, net.Tagger can employ these resources free of any reimbursement or compensation as long as any use is properly credited. An explicit goal of the project is the eventual integration of net.Tagger's data into the OSM community. Further, because the OSM community represents a large population segment of users who have similar motivations to the desired net.Tagger user community, e.g., individuals who voluntarily annotate maps, bidirectional interaction between net.Tagger and OSM is a potential means of furthering net.Tagger's goals. Such integration could be accomplished by importing verified net.Tagger data into the OSM dataset. OSM emphasizes above-ground features that can be verified by other mappers as part of its implementation philosophy, with no real means to record virtualized inferences of below-ground networks [57]. However, the street-level infrastructure indicators from net.Tagger can be recorded in OSM much like other street level OSM features such as bike racks or utility poles. Importing part or all of the eventual net.Tagger dataset into OSM is not without its potential disadvantages, and would only happen after a careful cost-benefit analysis. Any import could only take place after interacting with and gaining approval from the OSM Import Mailing List [58] to ensure that the bulk data met OSM standards and was appropriately categorized.

5.2.3 Native Renders

Currently, the only means to render tag data in a map overlay is through the app's Google Maps API. The Google Maps API was chosen as an expedient way to meet the project's time constraints. Although useful for prototyping, long-term reliance on a proprietary mapping API conflicts with several of net.Tagger's core objectives. net.Tagger aims to provide map renders on multiple platforms, including Android, IOS, and web browsers. Additionally, net.Tagger seeks to maintain as much compatibility with OSM as possible to permit use of and possible future integration with the OSM dataset. Finally, most members of net.Tagger's target user community are associated with open source projects and initiatives that emphasize information sharing and openness of data and methods. Considering these factors, migrating map renders to an open source, OSM compatible approach is a logical next step for both web and app displays. Fortunately, the OSM software stack meets all of these criteria. Although there is no one standard OSM approach to rendering and serving map tiles, a standard community approach uses an open source rendering software known as Mapnik [59] [60] in combination with helper packages to pull data from a PostGIS database, overlay it onto an existing GIS dataset (such as the OSM planet file), and serve the resulting map tiles via an Apache web server. Various OSM sub-communities provide documentation of their setups to assist others in deploying map servers using free, open source software. Various toolkits also exist to directly integrate OSM data into apps. One example is OSMDroid [61], an open source toolkit using OSM data as a direct replacement for most GoogleMaps API features. This would permit a straightforward port of the net.Tagger app from GoogleMaps to OSM based displays without requiring extensive code rewrites. The net.Tagger project can incorporate these resources as part of its expanded web and app presence.

5.3 Data Analysis

While much of this thesis covers net.Tagger's crowdsourcing implementation, the core goal of the project remains analyzing and drawing useful physical network topology inferences. Before useful analysis can take place, collected data must be initially categorized and vetted. A key part of this process is extracting information from submission images and augmenting the user's form data inferences. However, the anticipated volume of data implies that manual inspection by the small project team is not possible. Several possibilities exist to automate or outsource this process.

5.3.1 Image Recognition

Although image recognition technology has limitations, it still represents a potential means to identify net.Tagger's targets. Many of the indicators in Section 2.5 have distinct shapes such as circles (manhole covers) and rectangles (handholes), or color (PMS 144 Orange).

Image recognition software could theoretically search for these predetermined shapes and colors in user images and check them against what the user identified as the find. Depending on the image quality and camera perspective, markings and text in images could potentially be analyzed with Optical Character Recognition (OCR) software as well, however, human-based verification will also play a large role. More complex shapes such as cell towers and buildings may not lend themselves to automated cataloguing, due to their lack of a generalized shape or intentional obfuscation, as discussed in Section 2.5.6. However, all other infrastructure indicators possess a specific shape that can be target with information recognition software.

5.3.2 Mechanical Turk

To extract more detailed information from images, net.Tagger could integrate with Amazon's Mechanical Turk service [62]. Mechanical Turk is a crowdsourced Amazon Web Service (AWS) allowing individuals, researchers, or businesses to submit Human Intelligence Tasks (HITs), small chores that are difficult to complete via computer but easily accomplished by a human being. Volunteers perform the tasks and receive a small compensation for each HIT, usually on the order of a few cents. Mechanical Turk lends itself well to image processing, particularly matching patterns or extracting text. These capabilities could be employed to verify images such as the previously mentioned cell towers and buildings. A sample HIT might involve presenting an image that a net.Tagger user categorized as a Level3 Telecommunications building, then asking the Mechanical Turk user questions such as "Is this picture of a building? What company names are present?" Mechanical Turk could also be used to supplement automated image recognition. For example, orange street markings frequently contain descriptive labels written freehand in street paint that are far less legible than stamped manhole inscriptions. If image recognition software detects the PMS 144 color in a user submission, the image could be redirected to Mechanical Turk to ask if any phrases exist in the picture.

5.4 User Incentives

The success of any crowdsourcing project relies on a simple principle: the project must provide its users with reasons motivating them to join, contribute, and continue participating long enough to provide useful data. Incentives can take many forms, including monetary, prestige, or conditional access to an asset. Depending on the resources available to a project, multiple forms of incentives can be combined to target a larger potential user base.

5.4.1 Leaderboard

The planned incentive net.Tagger will incorporate into its initial large-scale deployment involves recognizing users based on the quantity, quality, and type of their submissions. These rankings will be displayed in an online "leaderboard" displaying users according to their tagging accomplishments. A key advantage of such a system is that net.Tagger administrators can assign points (or negative points) to different types of actions that factor into a user's ranking score. Possible point strategies for different categories of submission include:

- Submitting an original tag with an accompanying image and user comments. This would be worth the maximum number of points, as it provides not only the standard submission data, but a means of verification. For example, if a user selects one infrastructure type from the app UI, but enters comments about a different type, researchers can assign a lower probability that the submission is accurate. An image provides even better verification ability, where researchers can clearly see if a user inferred correct information about a submission.

- Submitting an original tag without an image or comments. In order to account for users with constraints on their time or phone data plans, net.Tagger provides the ability to submit tags containing only app form data and GPS sensor information. These submissions are still useful, particularly if verified through multiple users tagging the same find. However, they provide less data than a full submission, and would be worth fewer points.

- A bonus for submitting an especially valuable tag. A unique feature of net.Tagger is its ability to gather data about infrastructure indicators that only exist temporarily, primarily orange street markings that eventually fade and wash away (section 2.5.1). These markings provide some of the best data, including the streetwise orientation of the infrastructure. Because the markings exist for a much shorter time than more permanent infrastructure such as manhole covers, any indicated provider name is more likely to be current and accurate. The leaderboard algorithm can provide a point bonus for submission and verification of temporary markings, encouraging users to seek them out before they disappear.

- Verifying another user's submission. To increase the validity of research data, users can be prompted to seek out and verify other submissions. This feature could not be implemented until the enhanced map display (5.1.5) is implemented. A verification feature could be presented to users as a means for newer users to gain early points.

The verification feature introduces new error handling abilities, but must be handled carefully to avoid unintended consequences. Allowing users to essentially "challenge" submissions made by others if they cannot replicate the same results might provide an incentive to submit false tags to earn points for themselves while subtracting points from the original tagger. Unethical users trying to attain and stay at the top of the leaderboard could easily take advantage of verifications. Even discounting the potential effects of user misconduct, other situations might produce negative results as well. Because of their non-permanency, orange street markings disappear after a relatively short amount of time, and a user attempting to verify them weeks or months after the original tag could find nothing and submit a challenge even though the initial tag was correct. The variable accuracy of smartphone GPS units means that a tagged item does not exist where the tag lat/long indicates, but somewhere in a circle with a radius equal to the GPS error. In dense urban areas with high concentrations of infrastructure indicators, a verifying user might go to a tagged location, mistake one infrastructure indicator for another, and erroneously verify or challenge the wrong indicator. The verification process will require careful planning to avoid exploitation or inadvertantly introducing additional errors into the net.Tagger dataset.

In addition to a web-based leaderboard, the app will eventually have a local leaderboard of its own. The online leaderboard has the advantage of immediate access to the net.Tagger database, making calculation and display of the entire user community straightforward. Pushing out these results to the distributed network of user smartphones, however, is less simple. To compromise, each smartphone's leaderboard might display a smaller subset of results. This can be automated simply, with each app instance requesting updated results from the net.Tagger server once per day and receiving the ranked top ten as well as the standing of the user associated with the specific instance.

To further encourage competition, the user community can be permitted to form teams ranging from small groups of peers to entire countries. Displaying leaderboard rankings by country can be done with minimal extra effort because the information is included in each user profile. Allowing users to form additional groups would foster collaboration on a smaller scale.

5.4.2 Micropayments

Much like Amazon's Mechanical Turk, users could be paid a small amount in money or some form of credit. This feature would not be feasible without project sponsorship, and would thus be reserved for more mature releases. Because users might be tempted to submit false data to gain monetary rewards, delaying this feature would also allow fine-tuning of the verification process to better identify and prevent user fraud. Providing monetary compensation for all users and all submissions could easily lead to fraud, with users submitting fake tags in order to artificially boost numbers. Users would likely be required to undergo additional registration or vetting before becoming eligible to receive compensation. They might be initially required to to submit a certain number of verified tags, and only begin recieving compensating after passing a predetermined threshold. Even though this increases the administrative burden on project administrators, only a small number of users would likely qualify for this feature. As OSM demonstrates [32], the majority of high quality submissions would likely come from only a few percent of project participants. In order to increase the difficulty of faking a tag, compensation would be limited to submissions including images.

5.4.3 Dataset

Like OSM, net.Tagger's potential users exist on a spectrum, from casual users participating as a novelty to more dedicated, enthusiastic users with technical backgrounds employed in related areas of research or academia. Less invested users are unlikely to be interested in the accumulated project data beyond viewing maps of their findings. However, users working in similar research areas might desire access to portions of the net.Tagger dataset. Where micropayments would target high-performing individual users, access to part of net.Tagger's dataset would be an incentive aimed at research groups or similar entities providing some benefit to net.Tagger through established relationships. Much like micropayments and exporting data to the OSM project, providing other researchers access to the net.Tagger dataset would not be implemented until the project matures, in contrast to leaderboard implementation, which is of immediate interest.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] E. Çetinkaya, M. Alenazi, A. Peck, J. Rohrer, and J. Sterbenz, "Multilevel resilience analysis of transportation and communication networks," *Telecommunication Systems*, vol. 60, no. 4, pp. 515–537, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11235-015-9991-y

[2] R. Durairajan, S. Ghosh, X. Tang, P. Barford, and B. Eriksson, "Internet atlas: A geographic database of the Internet," in *Proceedings of the 5th ACM Workshop on HotPlanet*, ser. HotPlanet '13. New York, NY, USA: ACM, 2013, pp. 15–20. [Online]. Available: http://doi.acm.org/10.1145/2491159.2491170

[3] R. Singel. (2008, January). Fiber optic cable cuts isolate millions from Internet, future cuts likely. Wired Magazine. [Online]. Available: http://www.wired.com/2008/01/fiber-optic-cab/

[4] J. Hawkinson and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)," RFC 1930 (Best Current Practice), Internet Engineering Task Force, Mar. 1996, updated by RFCs 6996, 7300. [Online]. Available: http://www.ietf.org/rfc/rfc1930.txt

[5] P. FARATIN, D. CLARK, S. BAUER, W. LEHR, P. GILMORE, and A. BERGER, "The growing complexity of Internet interconnection," *Communications & Strategies*, vol. 1, no. 72, pp. 51–72, 4th quart 2008. [Online]. Available: https://www.akamai.com/us/en/multimedia/documents/technical-publication/the-growing-complexity-of-internet-interconnection-technical-publication.pdf

[6] R. Durairajan, P. Barford, J. Sommers, and W. Willinger, "Intertubes: A study of the us long-haul fiber-optic infrastructure," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 565–578, Aug. 2015. [Online]. Available: http://doi.acm.org/10.1145/2829988.2787499

[7] A. Taube. (2013, Jun). Sprint settles cable right-of-way suit. Law360. [Online]. Available: http://www.law360.com/articles/453369/sprint-settles-cable-right-of-way-suit

[8] If you own or owned land under or next to railroad rights of way where fiber-optic cable was installed, you could receive money from a class action settlement. FiberOpticSettlements. [Online]. Available: https://fiberopticsettlements.com/rhodeisland/Portals/0/Documents/RI%20ROW%20Notice.pdf

[9] D. Tan, A. Grieco, and Y. Fainman, "Towards 100 channel dense wavelength division multiplexing with 100ghz spacing on silicon," *Optics express*, vol. 22, no. 9, pp. 10 408–10 415, 2014.

[10] (2012). Executive order – accelerating broadband infrastructure deployment, EO 13616. https://www.whitehouse.gov/the-press-office/2012/06/14/executive-order-accelerating-broadband-infrastructure-deployment. Washington DC.

[11] (2012). Executive order on accelerating broadband infrastructure deployment. Federal Highway Administration. [Online]. Available: http://www.fhwa.dot.gov/policy/otps/exeorder.cfm

[12] H.r.3805 - broadband conduit deployment act of 2015. [Online]. Available: https://www.congress.gov/bill/114th-congress/house-bill/3805/titles

[13] D. McCabe. (2015, Oct). 'dig once' eyed for broadband expansion. [Online]. Available: http://thehill.com/policy/technology/257981-dig-once-eyed-for-broadband-expansion

[14] Broadband opportunity council final report. [Online]. Available: https://www.whitehouse.gov/sites/default/files/broadband_opportunity_council_report_final.pdf

[15] A. Blum, *Tubes: A Journey to the Center of the Internet*. New York City, NY: Ecco, 2013.

[16] (2001, Jul). National traffic safety board railroad accident brief. National Traffic Safety Board. [Online]. Available: http://www.ntsb.gov/investigations/AccidentReports/Reports/RAB0408.pdf

[17] N. J. Victory. (2006, Jun). Report and recommendations of the independent panel reviewing the impact of hurricane katrina on communications networks. Federal Communications Commission. [Online]. Available: https://transition.fcc.gov/pshs/docs/advisory/hkip/karrp.pdf

[18] F. Holzhauer, "Ip geolocation," TU Berlin, Tech. Rep., 2007. [Online]. Available: http://www.net.t-labs.tu-berlin.de/teaching/ss07/IM_seminar/aa_b1.pdf

[19] S. Siwpersad, B. Gueye, and S. Uhlig, "Assessing the geographic resolution of exhaustive tabulation for geolocating Internet hosts," in *Passive and Active Network Measurement*, ser. Lecture Notes in Computer Science, M. Claypool and S. Uhlig, Eds. Springer Berlin Heidelberg, 2008, vol. 4979, pp. 11–20. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-79232-1_2

[20] V. N. Padmanabhan and L. Subramanian, “An investigation of geographic mapping techniques for Internet hosts,” *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 173–185, Aug. 2001. [Online]. Available: http://doi.acm.org/10.1145/964723.383073

[21] B. Gueye, A. Ziviani, M. Crovella, and S. Fdida, “Constraint-based geolocation of Internet hosts,” in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’04. New York, NY, USA: ACM, 2004, pp. 288–293. [Online]. Available: http://doi.acm.org/10.1145/1028788.1028828

[22] E. Katz-Bassett, J. P. John, A. Krishnamurthy, D. Wetherall, T. Anderson, and Y. Chawathe, “Towards ip geolocation using delay and topology measurements,” in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’06. New York, NY, USA: ACM, 2006, pp. 71–84. [Online]. Available: http://doi.acm.org/10.1145/1177080.1177090

[23] B. Wong, I. Stoyanov, and E. G. Sirer, “Octant: A comprehensive framework for the geolocalization of Internet hosts,” in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, ser. NSDI’07. Berkeley, CA, USA: USENIX Association, 2007, pp. 23–23. [Online]. Available: http://dl.acm.org/citation.cfm?id=1973430.1973453

[24] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, “Measuring isp topologies with rocketfuel,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 2–16, Feb. 2004. [Online]. Available: http://dx.doi.org/10.1109/TNET.2003.822655

[25] (2015). Submarine cable map. TeleGeography. [Online]. Available: https://www.telegeography.com/telecom-maps/submarine-cable-map/

[26] Openstreetmap. Open Street Map Foundation. [Online]. Available: www.openstreetmap.org/

[27] Stats. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/Stats

[28] Editors. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/Editors

[29] (2015, May). Copyright easter eggs. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/Copyright_Easter_Eggs

[30] Faq. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/FAQ

[31] P. N. Marco Helbich, Christof Amelunxen. (2010). Comparative spatial analysis of positional accuracy of openstreetmap and proprietary geodata. [Online]. Available: http://koenigstuhl.geog.uni-heidelberg.de/publications/2010/Helbich/Helbich_etal_AGILE2011.pdf

[32] J. J. Arsanjani, C. Barron, M. Bakillah, and M. Helbich, "Assessing the quality of openstreetmap contributors together with their contributions," in *AGILE 2013*, 2013.

[33] Import/catalog. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/Import/Catalogue

[34] Humanitarian openstreetmap team: Haiti. Humanitarian OpenStreetMap Team. [Online]. Available: https://hotosm.org/projects/haiti-2

[35] (2016). Portolan network sensing architecture. Portolan Project. [Online]. Available: http://portolanproject.iit.cnr.it/

[36] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based Crowdsourcing for Network Monitoring: Opportunities, Challenges, and a Case Study," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 106–113, Jan. 2014. [Online]. Available: http://dx.doi.org/10.1109/mcom.2014.6710071

[37] A. Faggiani, E. Gregori, L. Lenzini, S. Mainardi, and A. Vecchio, "On the feasibility of measuring the Internet through smartphone-based crowdsourcing," in *WiOpt*, 2012, pp. 318–323.

[38] (1999, Apr). Apwa uniform color code. American Public Works Association. [Online]. Available: https://www.apwa.net/content/library/colorcc.pdf

[39] What do utility paint markings mean? Huntsville Utilities. [Online]. Available: https://www.hsvutil.org/ac/wp-content/uploads/2013/10/APWA_ColorCode.jpg

[40] Waymarking.Com. (2009, Mar). Evergreen tree cell tower. [Online]. Available: http://img.groundspeak.com/waymarking/f3283d04-d969-4bf4-9e51-dd9435545274.jpg

[41] (2012, Mar). Cities divided over cellphone towers. The Orange County Register. [Online]. Available: http://www.ocregister.com/articles/park-342841-cell-wireless.html

[42] Disguised cell towers. Waymarking.com. [Online]. Available: http://www.waymarking.com/cat/details.aspx?f=1&guid=5df351c0-98ea-4b8c-9a84-844f67beb552&st=2. Accessed: 2016-02-09.

[43] android.location. Android Developers. [Online]. Available: http://developer.android.com/reference/android/location/package-summary.html

[44] Making your app location aware. Android Developers. [Online]. Available: http://developer.android.com/training/location/index.html

[45] Sensors overview. Android Developers. [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html

[46] Mobile app. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Mobile_app

[47] A. Burnap, Y. Ren, R. Gerth, G. Papazoglou, R. Gonzalez, and P. Y. Papalambros, "When crowdsourcing fails: A study of expertise on crowdsourced design evaluation," *Journal of Mechanical Design*, vol. 137, no. 3, p. 031101, 2015.

[48] Zip code. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/ZIP_code

[49] (2016, Mar). Google maps geocoding api. Google Developers. [Online]. Available: https://developers.google.com/maps/documentation/geocoding/intro

[50] List of new york city manhole cover abbreviations. [Online]. Available: https://en.wikipedia.org/wiki/List_of_New_York_City_manhole_cover_abbreviations

[51] W. Boswel. (2013, Nov). Why users uninstall apps. Intel Corporation. [Online]. Available: https://software.intel.com/en-us/blogs/2013/11/14/why-users-uninstall-apps

[52] Position sensors. [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_position.html

[53] (2013, Nov). Core app quality. Android Developers. [Online]. Available: http://developer.android.com/distribute/essentials/quality/core.html

[54] L. Villapaz. (2015, Oct). Apple's iOS is still getting crushed by Android in the US. [Online]. Available: http://www.ibtimes.com/apples-ios-still-getting-crushed-android-us-2130868

[55] (2015, Jun). Php security cheat sheet. Open Web Application Security Project. [Online]. Available: https://www.owasp.org/index.php/PHP_Security_Cheat_Sheet

[56] (2016, Jun). Security tips. Apache Foundation. [Online]. Available: https://httpd.apache.org/docs/2.4/misc/security_tips.html

[57] (2013, Sep). Seeking to record fiber network topology using osm. Open Street Map Foundation. [Online]. Available: https://help.openstreetmap.org/questions/26483/seeking-to-record-fiber-network-topology-using-osm

[58] Imports. Open Street Map Foundation. [Online]. Available: https://lists.openstreetmap.org/listinfo/imports

[59] Mapnik. Open Street Map Foundation. [Online]. Available: http://wiki.openstreetmap.org/wiki/Mapnik

[60] (2016). Mapnik. Mapnik. [Online]. Available: http://mapnik.org/

[61] Osmdroid. Github. [Online]. Available: https://github.com/osmdroid/osmdroid

[62] (2016). Amazon mechanical turk. Amazon. [Online]. Available: https://www.mturk.com/mturk/welcome

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California